ARMSYS44B0-P 嵌入式系统开发板

使用说明书

（Ver. 20061011）

杭州立宇泰电子有限公司

http://www.hzlitai.com.cn

前　言

感谢您使用杭州立泰电子有限公司的 ARMSYS44B0-P 嵌入式系统开发板！

随着计算机技术逐渐渗透到各类电子产品当中，一种实用、高效的计算机系统——嵌入式系统不断展现出它独特的魅力。与桌面计算机不同，嵌入式计算机系统专门服务于特定需求，一般要求低成本、低功耗、轻型、高性能、高可靠性及可配置性。嵌入式系统日益广泛的应用也让人们看到了这项蕴涵的巨大的市场潜力。市场的需求带动了对技术人才的需求，今天，大批的技术人员和学者的目光都被吸引到嵌入式系统的设计与开发这门技术上。拥有一套好的嵌入式系统开发平台是每位技术人员的心愿，而 ARMSYS44B0-P 开发板就满足了您的需求。

ARMSYS44B0-P 开发板使用建议：

1. 了解 ARMSYS44B0-P 的硬件组成，你可以从附带光盘中提供电路原理图中找到更详细的电路资料。

2. 了解 ARMSYS44B0-P 提供的开发环境，并熟悉各种开发工具的使用。在本说明书的第四章有关于开发工具的使用例子,一步步指导用户如何使用各种开发工具。

3. 了解 ARMSYS44B0-P 提供的实例项目，我们提供 source（源代码包）其中包含了完整的，经过验证的实例代码。用户可以首先读懂这些代码，然后编译、仿真调试，观察实验现象。最后可根据自身需要修改这些代码来满足特定的应用需求。

4. 可以采用配套的书籍配合学习。我们采用的配套教材为《嵌入式系统设计与开发实例详解》。其中有实例分析、运行结果等。

5. 用户拿到开发板套件后请先仔细阅读该说明书，然后运行开发板测试程序，了解开发板的基本使用方法，同时验证开发板的各项功能。

目 录

第一章　ARMSYS44B0-P 开发板概述

ARMSYS44B0-P嵌入式系统开发板（基于S3C44B0X处理器）提供作工精良、资源丰富的硬件电路板，和完整全面、极具价值的源代码包和开发工具，是价廉物美、物超所值的ARM7 开发板。同时配有专业书籍的讲解，系统、完整、开放的技术内容，使您能够快速将ARM技术应用到新产品的开发中去。

ARMSYS44B0-P 嵌入式开发板采用 Samsung（三星）公司推出的 16/32 位 RISC 处理器 S3C44B0X（ARM7TDMI 内核），为手持设备和一般类型应用提供了高性价比和高性能的微控制器解决方案，为了使应用系统降低成本，S3C44B0X 提供了丰富的内置资源，包括：8KB cache，可选内部 SRAM，LCD 控制器（支持到 256 色 DSTN），带自动握手的 2 通道 UART（兼容 IrDA1.0 标准,具有 16-byte FIFO），4 通道 DMA，存储系统管理器（片选逻辑，FP/EDO/SDRAM 控制器），PWM 功能的 6 通道定时器，71 个通用 I/O 端口，实时时钟 RTC，8 通道 10 位 ADC，IIC-BUS 接口，同步 SIO 接口和 PLL 倍频器。

一、ARMSYS44B0-P 嵌入式开发板主要有以下部分组成

1、CPU：三星 S3C44B0X(ARM7TDMI 内核)微处理器；外部时钟为 8MHz，内部倍频至 64MHz。

2、Nor Flash：选用 16Mbit(2M×8Bit/1M×16bit) SST39VF160 或 AM29LV160，电路兼容 32Mbit(4M×8Bit/2M×16bit)SST39VF320 或 AM29LV320。

3、SDRAM：兼容 PC100/PC133，基本配置采用 64Mbit(1M×4Bank×16bit)HY57V641620，可选配 128Mbit(2M×4Bank×16bit)HY57V281620，或 256Mbit(4M×4Bank×16bit)HY57V561620。注意，由于电路兼容，实际电路板上可能选用其它型号的兼容型 SDRAM。

4、Nand Flash：基本配置采用 16M×8Bit 的 K9F2808,可选配 32M×8Bit 的 K9F5608 或 64M×8bit 的 K9F1208。电路兼容任何 8 位 Nandflash 芯片。

5、IIC 接口串行 EEPROM：AT24C04 4Kbit EEPROM。

6、UART 接口：2 通道 UART 接口，波特率高达 115200bps，具有 RS232 电平转换电路，可直接连接 PC 机。

7、LCD 和触摸屏接口：采用 26 针双排插针接口，其中包括 LCD 接口和触摸屏接口。LCD 接口利用 CPU 内部的 LCD 控制器扩展单色液晶屏或 256 色 STN/DSTN 型液晶屏（典型分辨率为 640×480、320×240），具有对比度调节电位器和液晶电源 5V/3.3V 选择跳线；触摸屏接口则为触摸屏控制器 ADS7843 接口设计。

8、IDE 及多功能扩展接口：可挂硬盘或 CompactFlash 卡，以及符合 IDE 扩展的外围设备，并配有 IDE 工作指示灯。

9、JTAG 接口：采用 20 针标准 JTAG 接口。

10、实时时钟 RTC：配有 3V（CR1220）纽扣电池，系统掉电后 RTC 仍能够保持工作。

11、复位电路和 4 路外部中断按键。

12、4 个可编程 LED 指示灯，1 个蜂鸣器。

13、PS/2 接口：可接 PS/2 键盘或鼠标。

14、以太网接口：采用 10M 以太网控制器 RTL8019，提供标准的 RJ45 接口座，可直接接入局域网。

15、USB 接口：采用 Cypress（赛普拉思）公司的 SL811HS 芯片具有主/从两种工作模式的 USB 控制器，遵循 USB1.1 规范。

16、音频接口：采用 PHILPS（菲利普）公司的 UDA1341 芯片通过 IIS 接口扩展出音频输入输出端口，可直接驱动耳机或其他音频设备。

17、SD Card/MMC 接口：通过 I/O 口扩展，兼容 SD 卡和 MMC 卡。

18、开发板供电：采用输出+5VDC 开关电源。（内正(+)，外负(-)接口）。

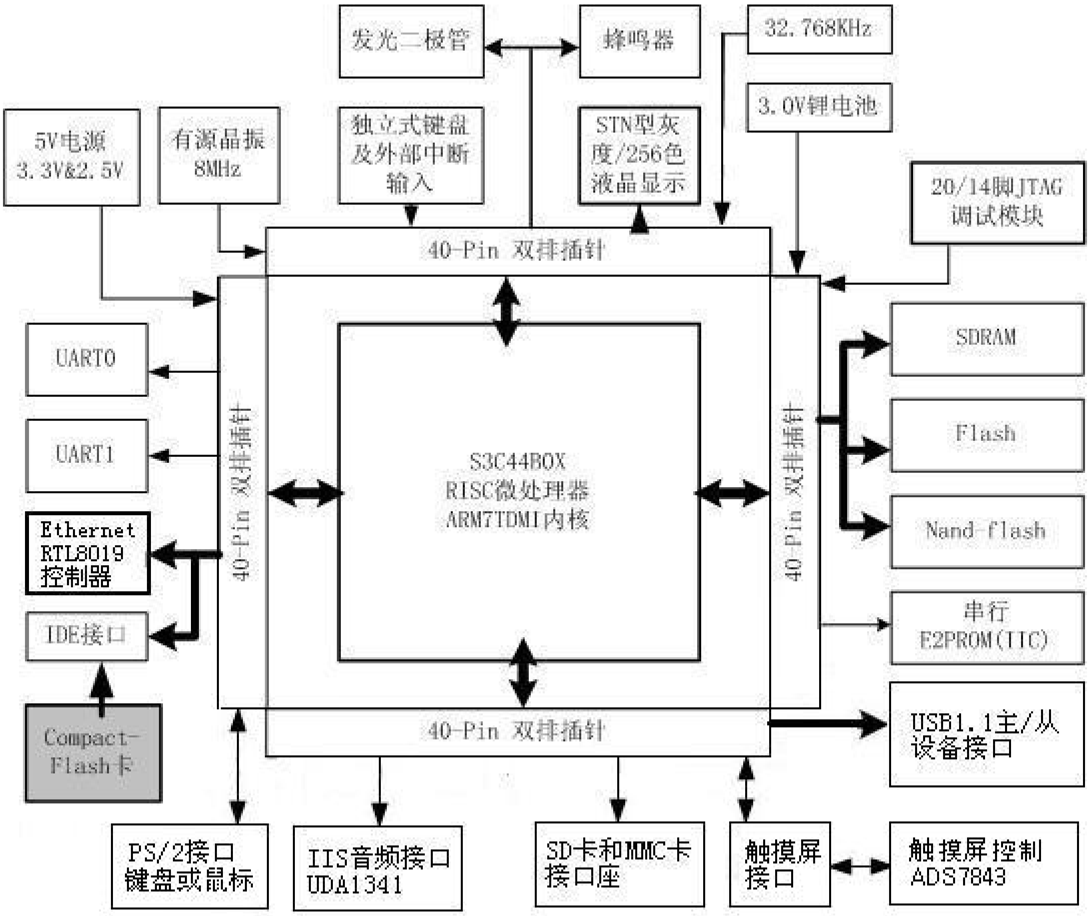


图 1—1 ARMSYS44B0-P 系统图

二、ARMSYS44B0-P 嵌入式开发板配套光盘资料介绍

1.2.1 “BIOS”文件夹

此文件夹中包含 ARMSYS44B0-P 开发板的启动工程文件(ADS1.2 工程) 及源程序。BIOS 主要提供代码下载功能和操作系统引导功能。

1.2.2 “source”文件夹

此文件夹中包含 ARM 工程模板和 24 个实例源程序。

其中 source 文件夹中包含 24 个非常实用的应用实例源码（按照字母排序），说明如下：

ADCtest	内置 A/D 转换器应用实例
ARPscan	APR 地址解析实例
ColorLCDtest	彩色液晶显示实例
Einttest	外部中断实例
Fat16test	fat16 文件系统应用实例
Flashtest	norflash 读写实例
Helloworld	Helloworld 程序
IICtest	IIC 接口运用实例
IIStest	IIS 音频接口播放音频实例
I/Otest	I/O 应用实例——矩阵式键盘扫描
NandFlashtest	非线性 flash 应用实例
Pingtest	以太网 ping 诊断应用实例
PS2KeyBoardtest	PS2 键盘接口应用实例
RTCtest	内置 RTC 应用实例
SD/MMCtest	SD 卡或 MMC 卡应用实例
SDRAMtest	SDRAM 存储器应用实例
SL811HostUSBtest	SL811 工作在主机方式（host）下的应用实例
Timertest	定时器应用实例
Touchtest	触摸屏应用实例
UARTtest	UART 串行口应用实例
ucOS_ex1	uC/OS-II 运用之一：多任务
ucOS_ex2	uC/OS-II 运用之二：任务间通讯
ucOS_ex3	uC/OS-II 运用之三：中断服务程序
UDPtest	UDP 数据报传输应用实例
template.mcp	工程模板

1.2.3 “uC/OS-II”文件夹

此文件夹中包含有 uC/OS-II 在 ARMSYS44B0-P 上的移植源程序包。

1.2.4、“uClinux”文件夹

此文件夹中包含有 uClinux 移植在 ARMSYS44B0-P 上的源码包、tftp 下载主机端程序、测试用映像文件和参考资料等。

1.2.5 “电路图”文件夹

此文件夹中包含有 ARMSYS44B0-P 开发板的原理图（pdf 文件格式）。

1.2.6 “开发工具”文件夹

此文件夹中包含有开发板使用到的工具软件或应用程序。其中包括：

1） 集成开发环境软件安装包（ADS1.2）。

2） “ARMJtagDebugFinal”文件夹中包含有 ARM7 和 ARM9 调试代理软件及驱动安装相关文件。

3）“programmer”文件夹中包含有快速烧录工程程序 ADS_programmer。烧录用二进制（bin）和 ELF 格式文件。

4）“flashpgm2.2.4”文件夹中包含有“flashpgm” 快速烧录工具安装包及配置文件 44B0.ocd。

5）“下载用的映像文件”文件夹中包含有所有恢复出厂配置的 bios,test,image 等一系列烧录用的二进制（bin）文件。

6） “tftp 下载工具”文件夹中包含有支持 linux 和 windows 的 ftp 下载工具。

7） “图形转 lcd 显示码工具”文件夹。

8） “sourceinsight35.rar”此压缩文件中为一个代码编辑器。

9） “dnw.exe”串口工具。

1.2.7 “芯片资料”文件夹

此文件夹中包含有开发板上采用的主要芯片及部件的数据资料。

1.2.8 ARMSYS44B0-P 开发板使用说明书

此文件为 ARMSYS44B0-P 开发板使用说明书（pdf 文件格式）。

光盘资料中各项内容更为细致的说明，参考每个目录下的 Readme.txt 文件。

三、ARMSYS44B0-P 开发板地址空间分配

S3C44B0X 处理器可以对 8 个 bank 进行寻址，每 bank 最大空间为 32M，为了使处理器对各个设备的访问互不干扰，我们将不同类的设备映射到不同的 bank 内。在 ARMSYS44B0-P 平台上，S3C44B0X 的 bank 空间的分配如图 1-2 所示。

图 1-2（a）是 ARMSYS44B0-P 的 BIOS 对程序空间和数据空间的分配。在程序空间 flash ROM 内（在主板上对应 2M 字节大小的 SST39VF160 器件），已经固化了一段启动系统并对系统进行初始化的程序——BIOS 程序。在图中，

可以看到 flashROM 存储器映射在了系统的 bank0 上，也就是说，系统上电时处理器即从 flashROM 的 0x00000000 地址处取得指令开始运行。这个地址上的 BIOS 程序完成了时钟设置初始化，中断矢量的定义，存储器的参数设置、堆栈地址定义等工作，这些设置对于系统正常启动是非常重要的。由于 flash ROM 是非易失性的存储器，因此程序就算掉电也不会丢失。但是如果由于某个误操作覆盖了 flashROM 中启动程序的内容，系统就将无法正常启动，这时就需要重新烧录 flashROM（参见相关章节）。烧录用的 44bapp.bin 文件和烧录工具都可以在光盘中的“开发工具\programmer ”目录下找到。

主板上的 SDRAM 器件映射在 bank6 上，也就是 0x0C000000 地址处。SDRAM 是易失性的可快速擦写的存储器，因此它通常作为系统的数据空间。

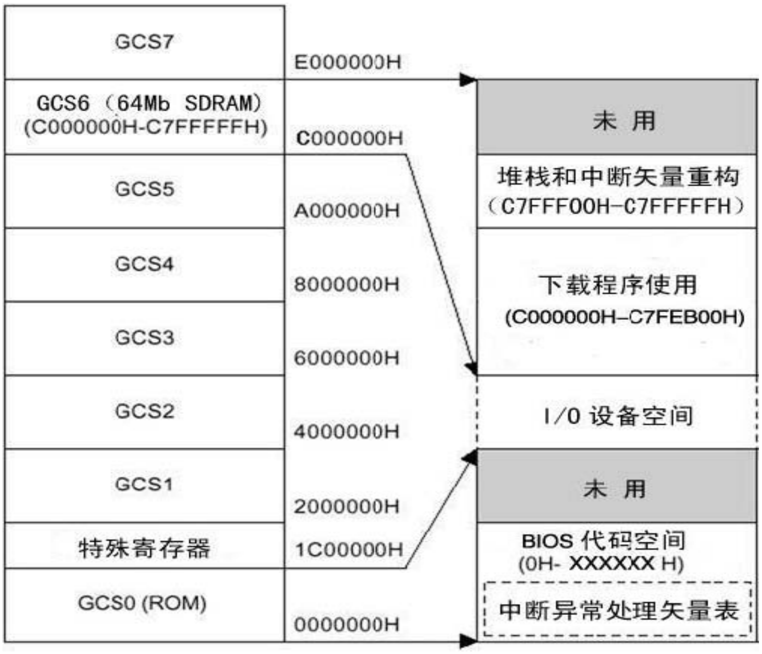
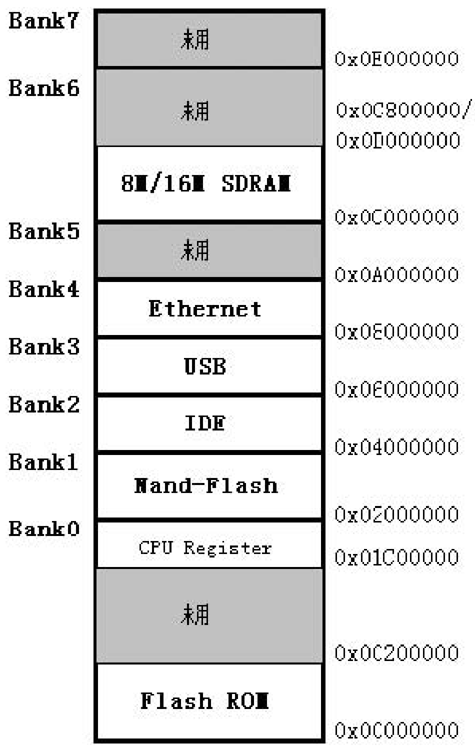


图 1－2(a) 程序和数据空间分配

如何设置系统的程序空间和数据空间呢？在开发环境（ADS）中通过对链接器的参数设置来完成。将只读基地址（程序空间的开始地址）RO Base 设置为 0x00000000，读写基地址（数据空间的开始地址）RW Base 设置为 0x0C5F0000，这样就完成了程序空间和数据空间的设置（参考 4.3.3 节）。在图 1-2（a）中，0xC000000～0xC7FEB00 作为下载的程序使用，就是指在仿真调试程序时，必须将 RO Base 地址设置为 0x0C008000 （0xC000000~0xC008000 存放了二级中断矢量，不可以被占用），这样程序空间定位在 SDRAM 中，程序就可以随时下载、更新了。这时，数据空间往往定义在更高地址的位置上，例如将 RW Base 设置为 0x0C5F0000，只要不与程序空间发生冲突即可。



系统片选和起始地址：

nGCS0 【 0x0000_0000 】： FLASH

（SST39VF160）

nGCS1【0x0200_0000】：NandFLASH（K9F2808）

nGCS2【0x0400_0000】：IDE/ATA

nGCS3【0x0600_0000】：SL811HST（USB HOST 或者是 USB DEVICE）

nGCS4【0x0800_0000】：RTL8019AS

nGCS5【0x0A00_0000】：SD Card/MMC

nGCS6【0x0C00_0000】：SDRAM（HY57V641620）

nGCS7【0x0E00_0000】：保留

图 1-2（b）中，显示了 ARMSYS44B0-P 的外部设备在处理器 bank 空间上的映射分布。Flash ROM 映射在 bank0 上，但它没有使用全部 bank0 的空间。Bank0 的高位空间地址，由 S3C44B0X 的内部特殊寄存器占用。Bank1，bank2，bank3，bank4，bank6 分别作为 Nand-flash，IDE， USB，以太网接口设备，SDRAM 的映射空间。与 Bank x 对应的片选引脚 nGCS x 将作为外部设备的使能脚使用（参考电路原理图）。

四、ARMSYS44B0-P 开发板接口及其他

1.4.1 系统基本参数设定

- BANK0 总线宽度为 16 bit。
- Little Endian (小端)模式。
- ARM 内核工作频率默认为 64MHz。
- 跳线 JP1 设定 USB SL811 的主从模式。
- 跳线 JP2 选择液晶供电电压。
- 跳线 JP3 用来复位同步 JTAG 和开发板。

1.4.2 开发板中断分配

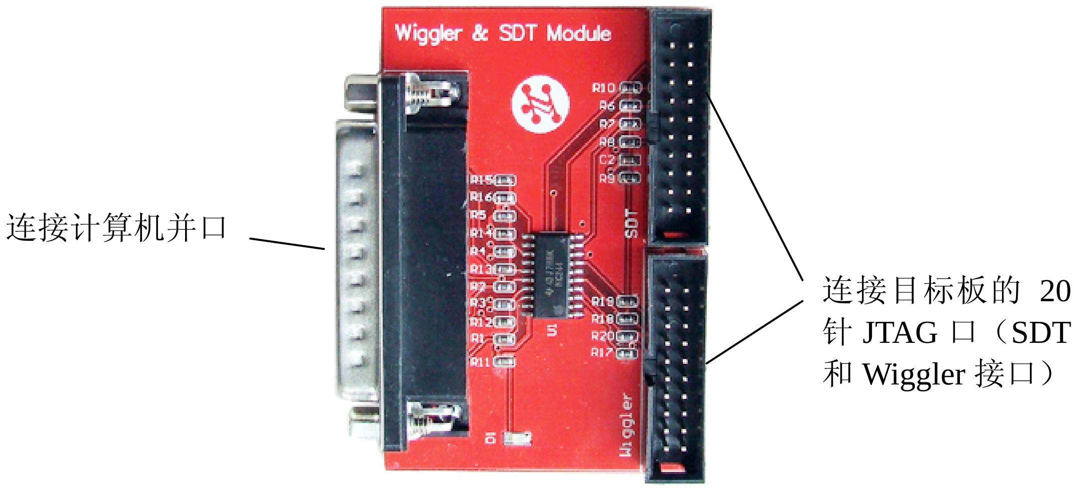
- EINT0：USB SL811HST
- EINT1：IDE/ATA
- EINT2：PS/2
- EINT3：RTL8019
- EINT4：SD/MMC (与按键 Exint4 共用)
- EINT5：SD/MMC (与按键 Exint5 共用)
- EINT6：按键 Exint6
- EINT7：触摸屏 TSP(与按键 Exint7 共用)

1.4.3 开发板对外接口与指示灯说明

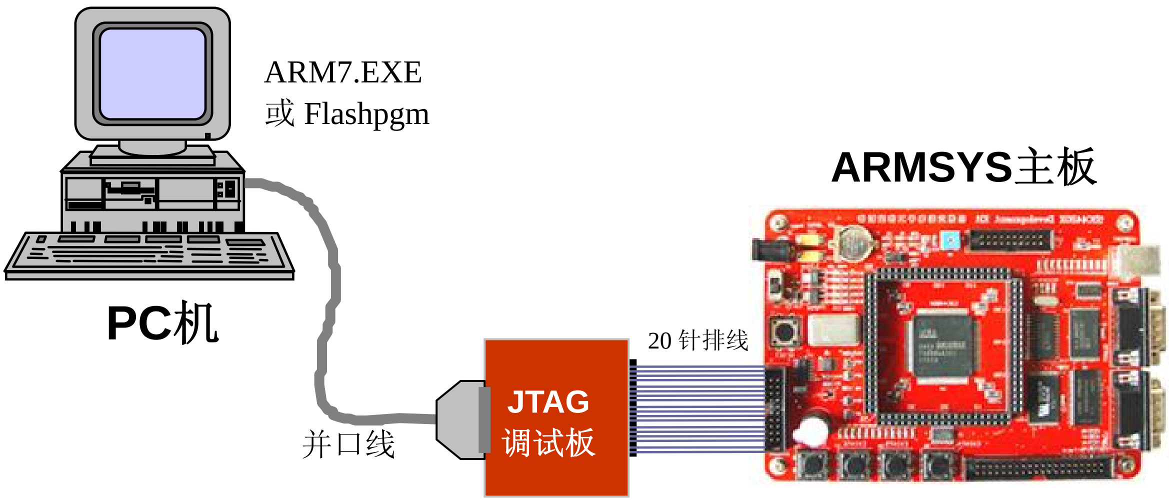
- LED1：5V 直流电源指示灯。
- LED2 和 LED3：RTL8019 工作状态指示灯。

- LED4：由 I/O 口 GPE7 控制的指示灯。
- LED5：由 I/O 口 GPE6 控制的指示灯。
- LED6：由 I/O 口 GPE5 控制的指示灯。
- LED7：由 I/O 口 GPE4 控制的指示灯。
- LED8：IDE/ATA 接口指示灯。
- 5V input：+5V 直流电源输入接口(内正外负)。
- SW1：5V 直流电源切换开关。
- RESET：开发板复位按键。
- JTAG20：20 针 JTAG 标准接口。
- LCD&TSP：26Pin 排线座，LCD 和触摸屏接口。
- USB-Host：USB—A 型接口，USB HOST 接口。
- USB-Slave：USB—B 型接口，USB DEVICE 接口。
- RJ45：以太网口插座。
- UART0：串口 0 接口座。
- UART1：串口 1 接口座。
- PS/2：PS/2 键盘或鼠标接口。
- EIDE Interface：40Pin 排线座，IDE/ATA 硬盘接口。
- Exint4：外部中断 4 按键。
- Exint5：外部中断 5 按键。
- Exint6：外部中断 6 按键。
- Exint7：外部中断 7 按键。
- SD/MMC：SD 卡座。
- LINE-OUT：IIS 立体声输出音频插座。
- MIC-IN：IIS 立体声话筒信号输入。
- J6：IIS 立体声输入音频接口。

五、JTAG 调试板的接口说明



JTAG 调试板的作用是程序下载和调试，一边采用并口线与 PC 机相连，一边 20 针排线与主板的连接，构成 ARMSYS 的调试系统。调试时的连接方式如下图所示：



使用 Wiggler 还是 SDT 口？

当用 ADS 或者 SDT 调试目标板时，可以采用 JTAG 调试板上 SDT 或 Wiggler 接口与目标板的 JTAG 接口相联接，只需要在调试代理软件上选择对应的接口即可。

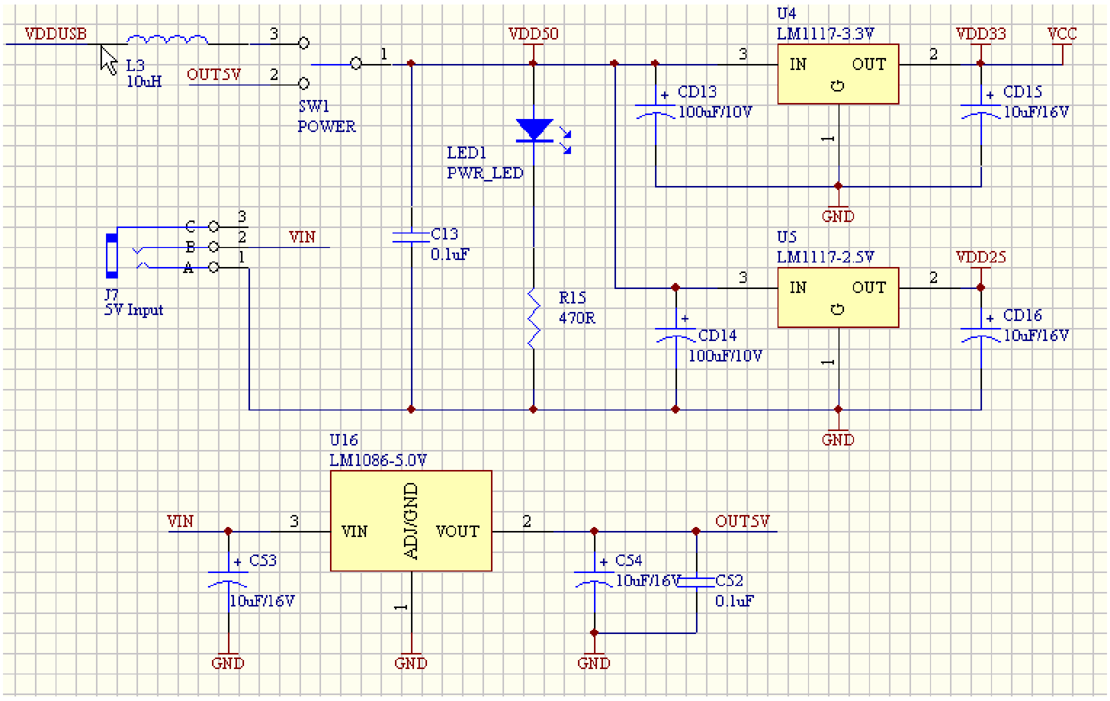
当用 Flashpgm 软件烧录程序时，必须采用 Wiggler 接口与目标板的 JTAG 接口相联接。

当用 Fluted 软件烧录程序时，必须采用 SDT 接口与目标板的 JTAG 接口相联结。

第二章 ARMSYS44B0-P 开发板电路描述

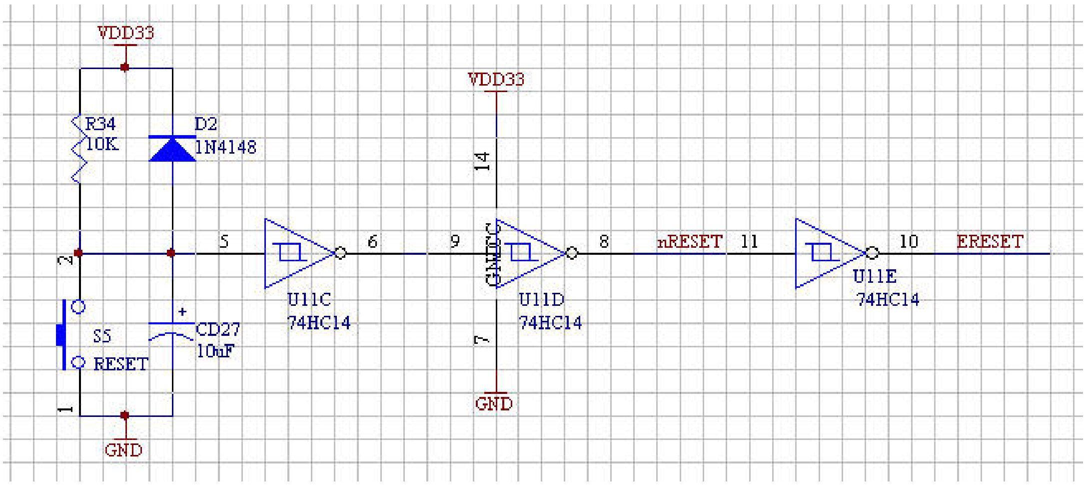
2.1 电源电路

开发板的供电可由外部直流稳压电源提供，为了提高供电的稳定性和可靠性，开发板采用集成线性稳压电源对 CPU 和外围电路供电。CPU 内核工作需要供给的 2.5V 电源采用 LM1117－2.5 产生。 CPU 的 IO 和外设工作所需要的 3.3V 电源是采用 LM1117－3.3 产生，相关电路如下：



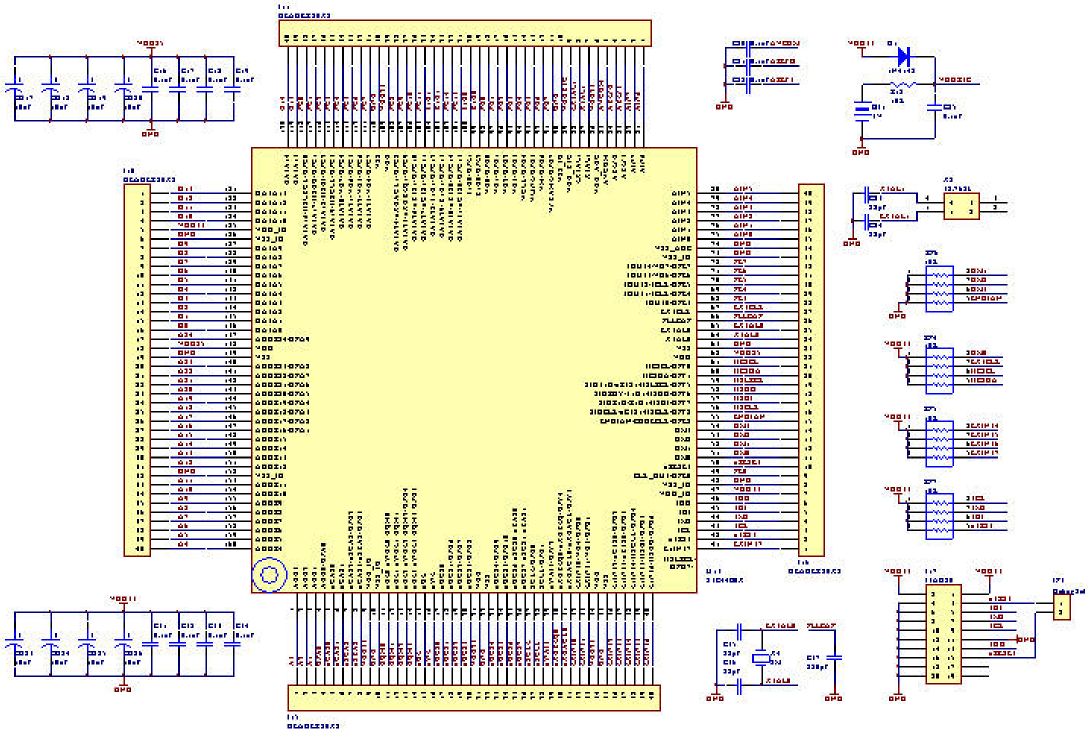
2.2 复位电路

采用较常用的 RC 加施密特反相器组成可靠的复位电路，如下所示：



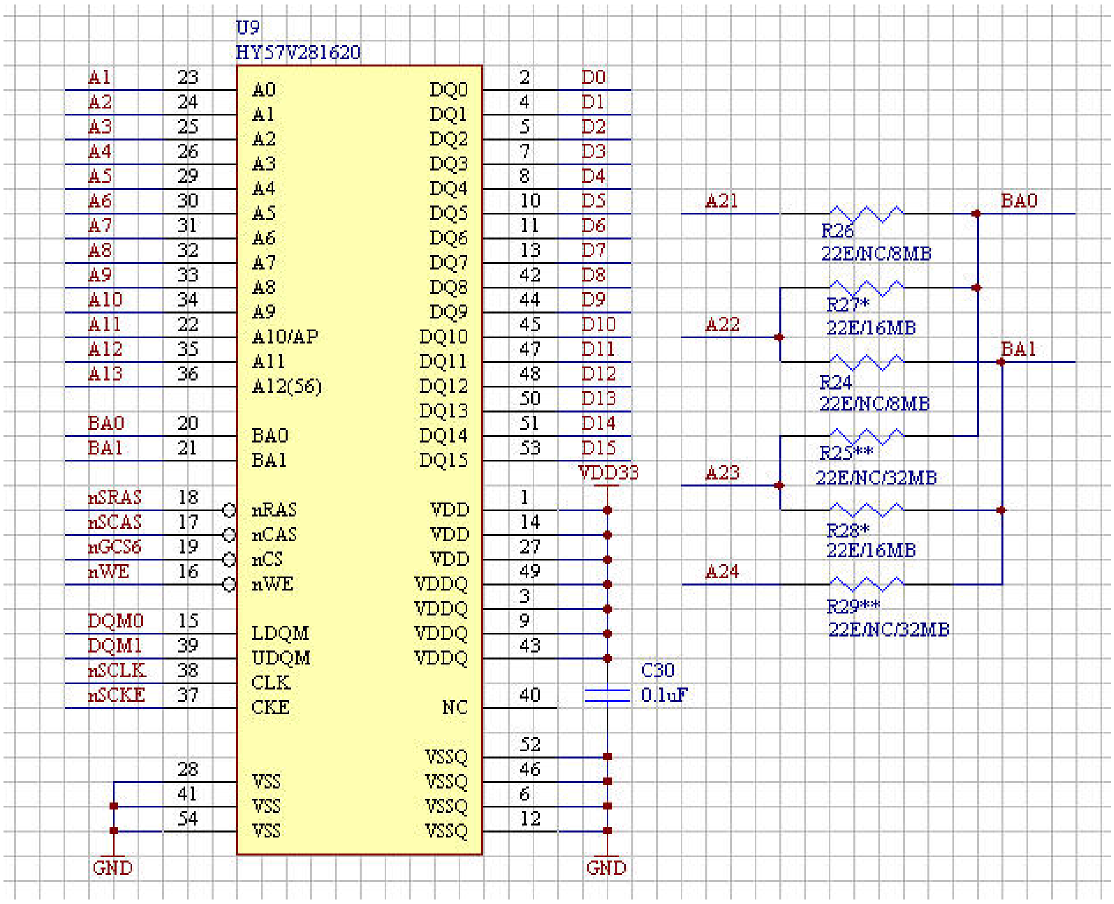
2.3　CPU 单元电路

CPU 单元电路采用开放式设计，用户可任意扩展。该电路包括有时钟电路、RTC 实时时钟调电电源电路和 JTAG 接口电路。如下所示：



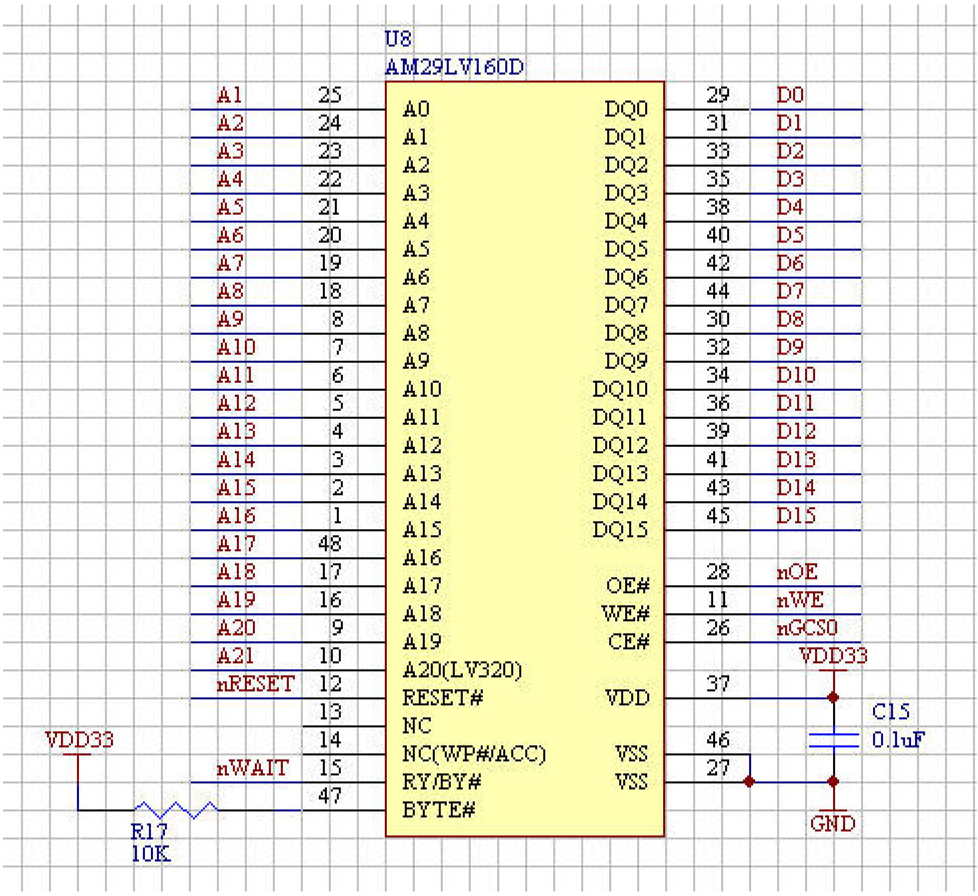
2.4　SDRAM 电路

SDRAM 电路采用了兼容性设计，考虑到用户的实际需要，支持 8M 字节、16M 字节和 32M 字节的 SDRAM，由三组六个电阻进行选择。如果是 8M 的 SDRAM（如 HY57V281620）则只焊接电阻 R26 和 R24，其他悬空不用。如果是 16M 的 SDRAM 则只焊接电阻 R27 和 R28，其他悬空不用。如果是 32M 的 SDRAM 则只焊接电阻 R25 和 R29，其他悬空不用。电路如下所示：



2.5 NorFlash 电路

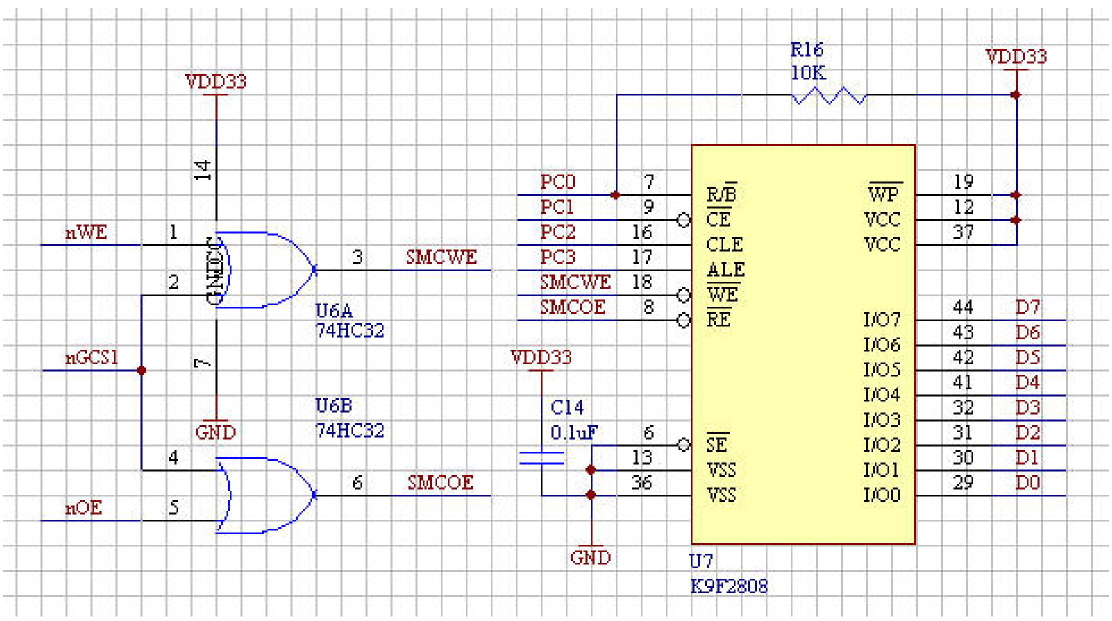
NorFlash 电路兼容 2M 字节和 4M 字节的 Flash，用户可以需要选择。电路如下所示：



由于采用电路兼容性的设计，实际板上采用芯片可能与电路中给出型号不同，用户也可以根据自身需要更换不同厂商的兼容型芯片。

2.6 NandFlash 电路

NandFlash 电路支持 8 位的 Nandflash，型号为 K9FXXX08，电路如下所示：

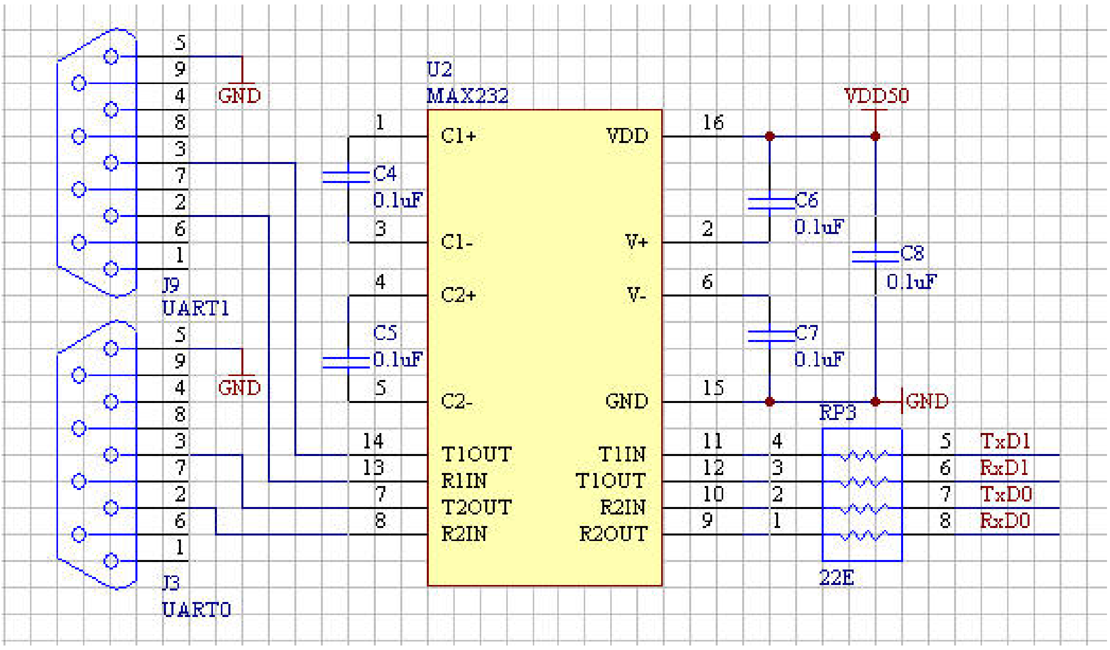


由于采用电路兼容性的设计，实际板上采用芯片可能与电路中给出型号不

同，用户也可以根据自身需要更换不同厂商的兼容型芯片。

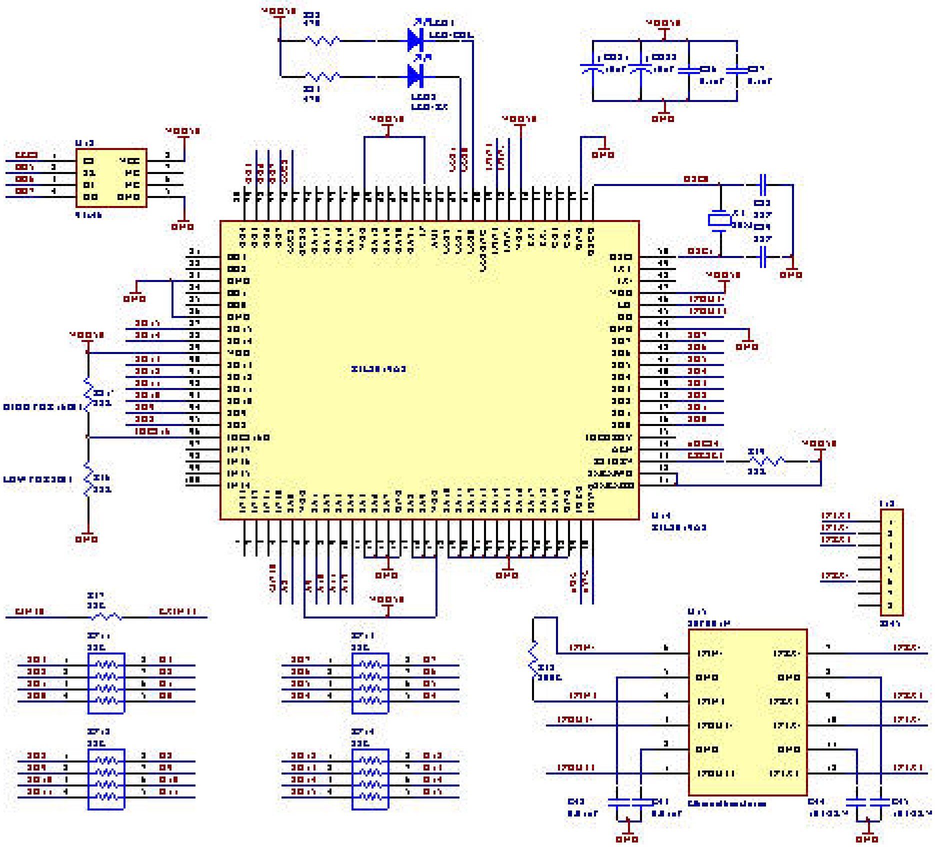
2.7 异步串行接口电路

该电路把 CPU 内部的两个串行口转换为标准的 RS232 接口。电路如下所示：



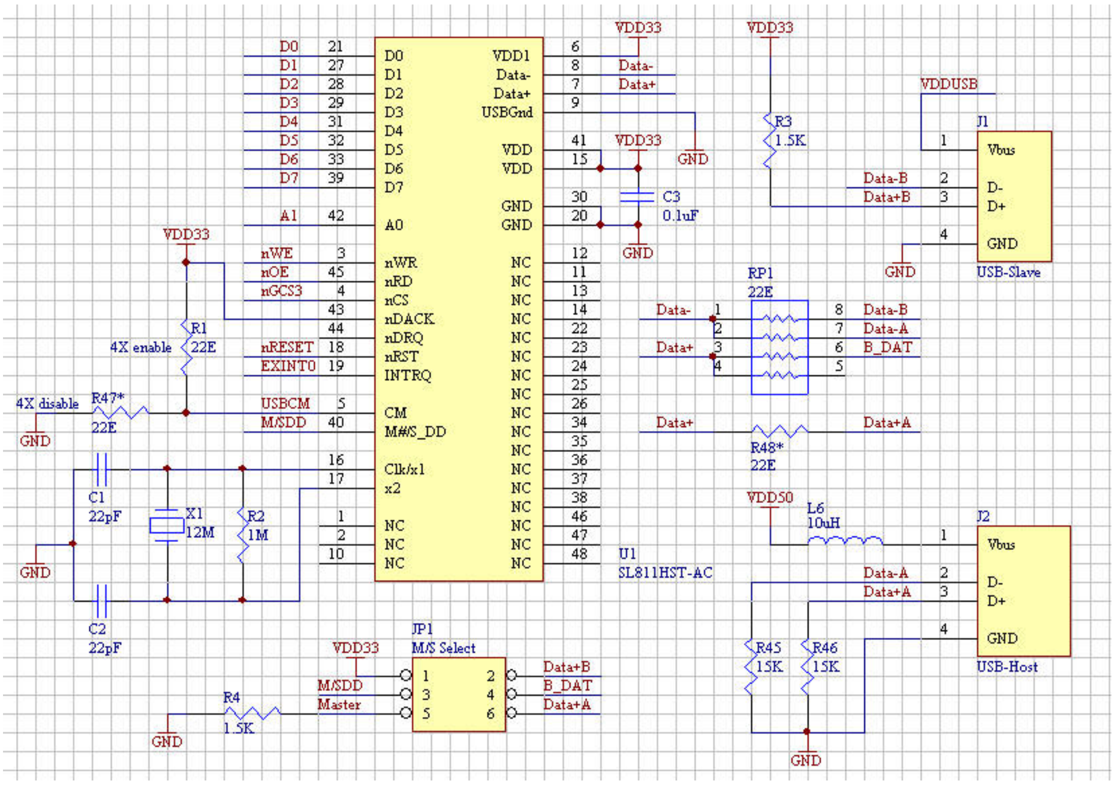
2.8 RTL8019 网络接口电路

该电路采用 10M 以太网接口控制器 RTL8019 为核心组成。与 CPU 的数据接口可 8 位或 16 位选择，默认数据位宽为 8 位。电路如下所示：



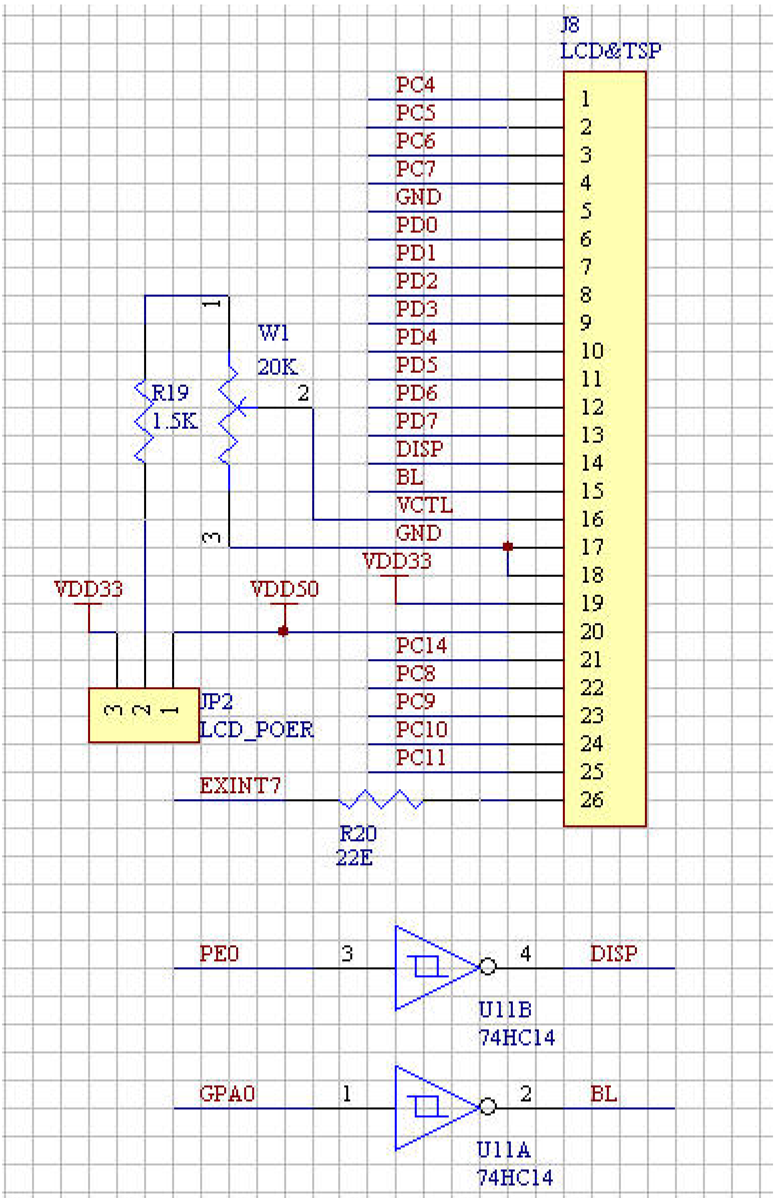
2.9 USB（SL811HST）接口电路

该电路为主从 USB 设计，通过 JP1 跳线选择。电路如下所示：



2.10 LCD 和触摸屏接口电路

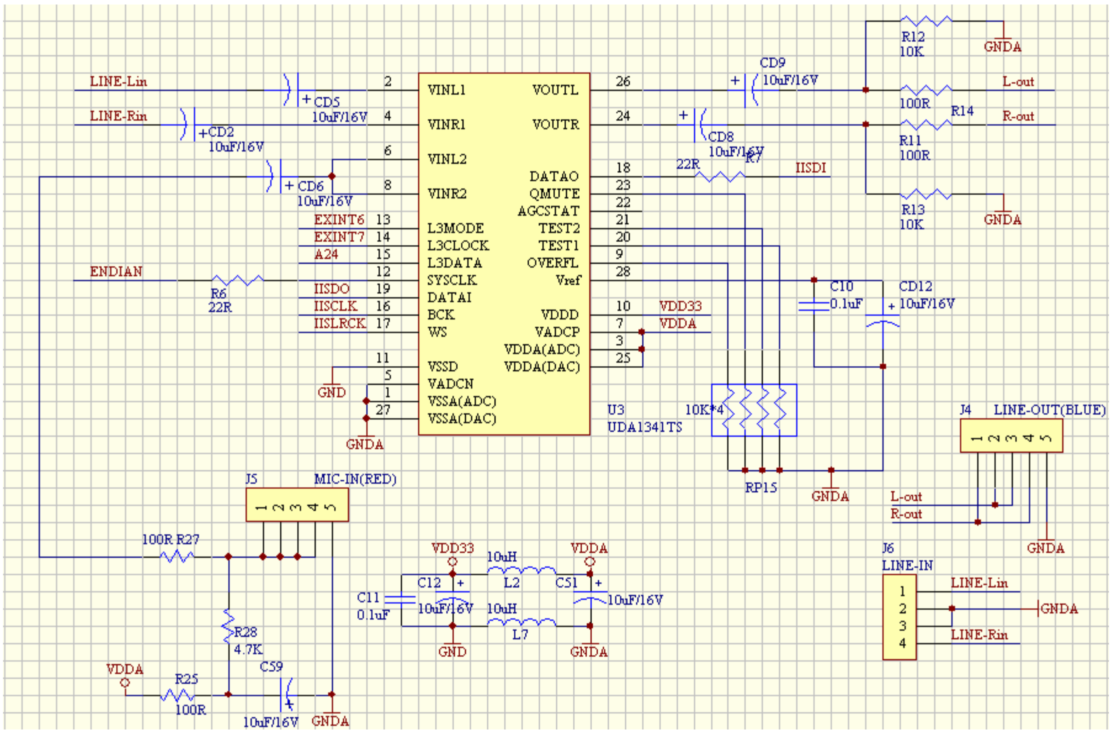
该电路将 LCD 的接口和触摸屏的接口整合在一起，通过排线可连接至液晶电路板，JP2 用来选择 LCD 液晶的工作电源。电路如下所示：



W1 用来调节液晶显示的对比度，PE0 口用来控制 STN 屏的 DISP-ON/OFF 口，GPA0 口用来开关背光。

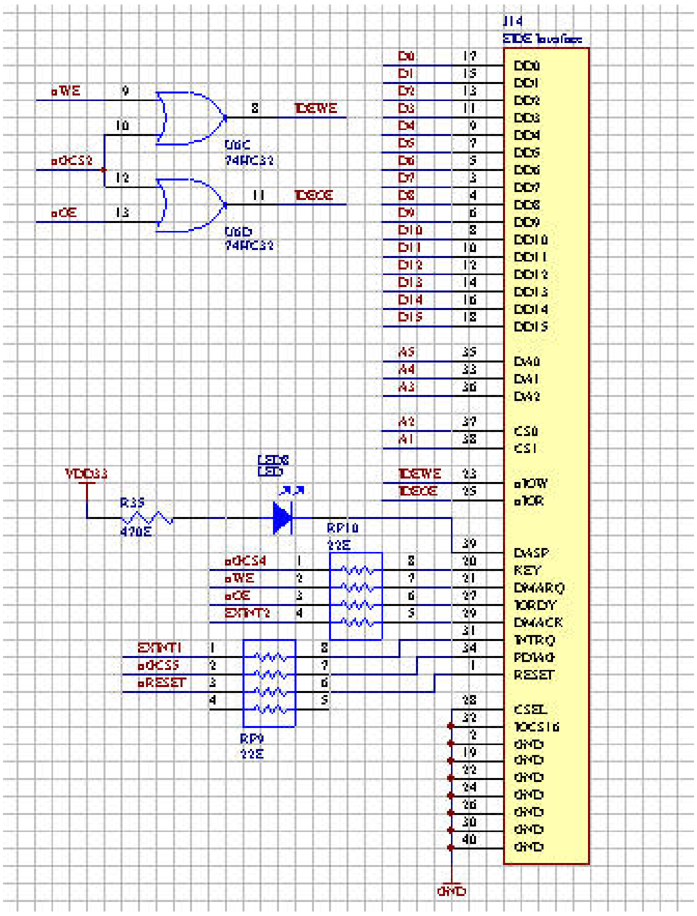
2.11 IIS 音频电路

该电路采用 UDA1341 专用音频芯片，可直接实现数字音频电路控制。电路如下所示：



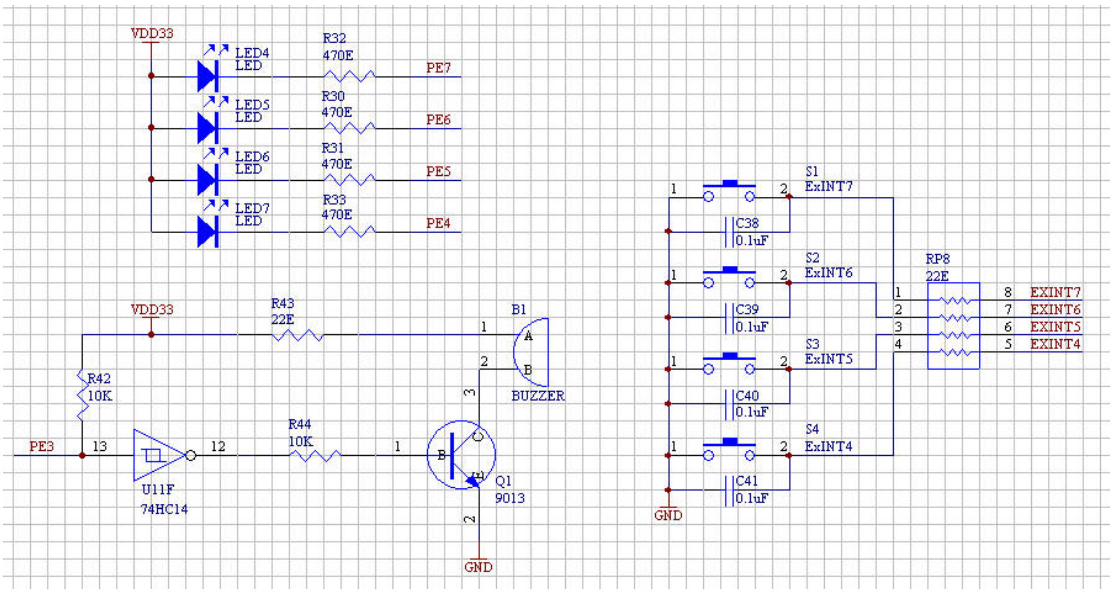
2.12 ATA(IDE)接口电路

ATA 接口电路入下所示，可挂接 IDE 硬盘或连接 CF 卡接口电路。



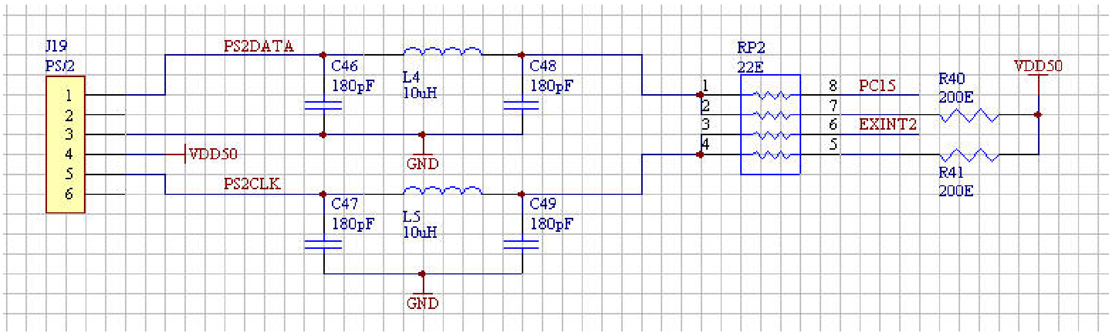
2.13 按键、蜂鸣器和 LED 电路

按键电路为 4 个外部中断的引出，分别对应外部中断 ExINT4、ExINT5、ExINT6 和 ExINT7。蜂鸣器和 LED 直接由 CPU 的 I/O 来控制。电路如下所示：



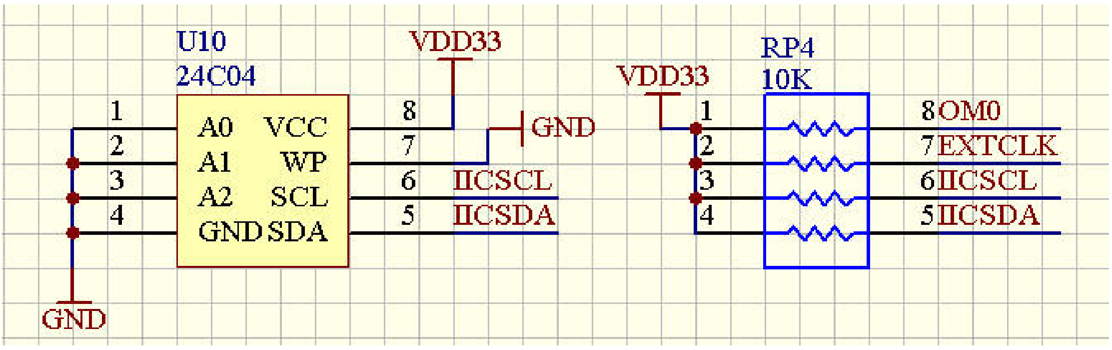
2.14 PS/2 接口电路

该电路为标准 PS/2 接口，可连接 PS/2 键盘或鼠标及其他设备。电路如下：



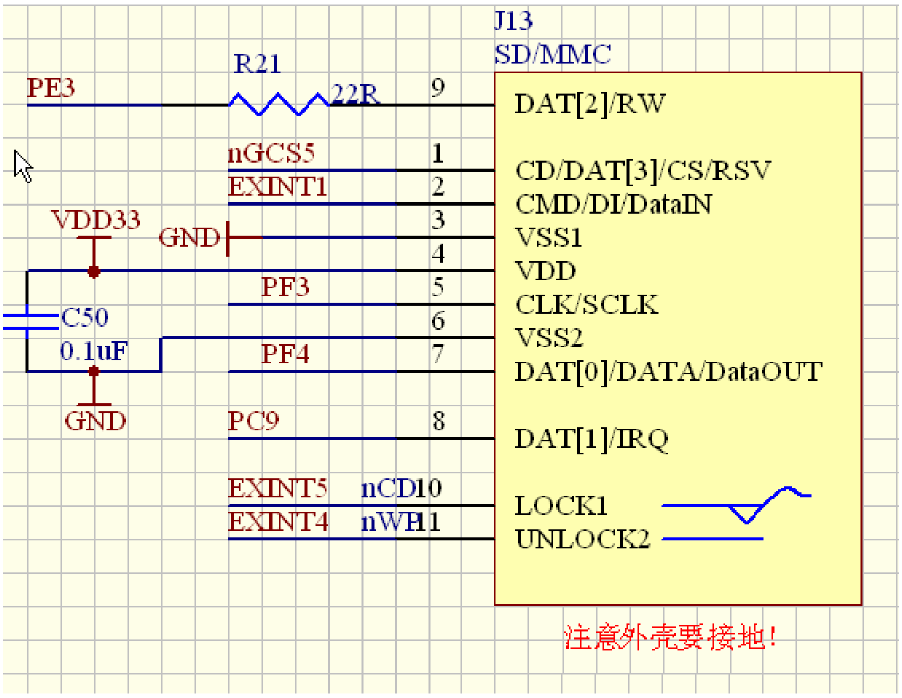
2.15 IIC 接口电路

IIC 电路采用存储器 AT24C04 芯片来实现，电路如下所示：



2.16 SD 卡接口电路

SD 卡电路设计为兼容 SD/MMC 方式，可以使用 SD 卡，也可用 MMC 卡，用户根据需要选择。电路如下所示：



第三章　BIOS 使用说明

3.1 ARMSYS44B0-P 开发板工作环境的建立

3.1.1 开发板工作电源

将配套的输出为+5VDC 的开关电源接至开发板，拨动电源开关 SW1 可使开发板上电开始工作，此时开发板上的红色电源指示 LED 点亮。

3.1.2 建立和设置超级终端

首先用开发板配套的串行电缆线连接计算机的串行口与开发板上的 UART0 串行口。

然后在计算机上点击[开始 | 程序 | 附件 | 通讯 | 超级终端]如下图。

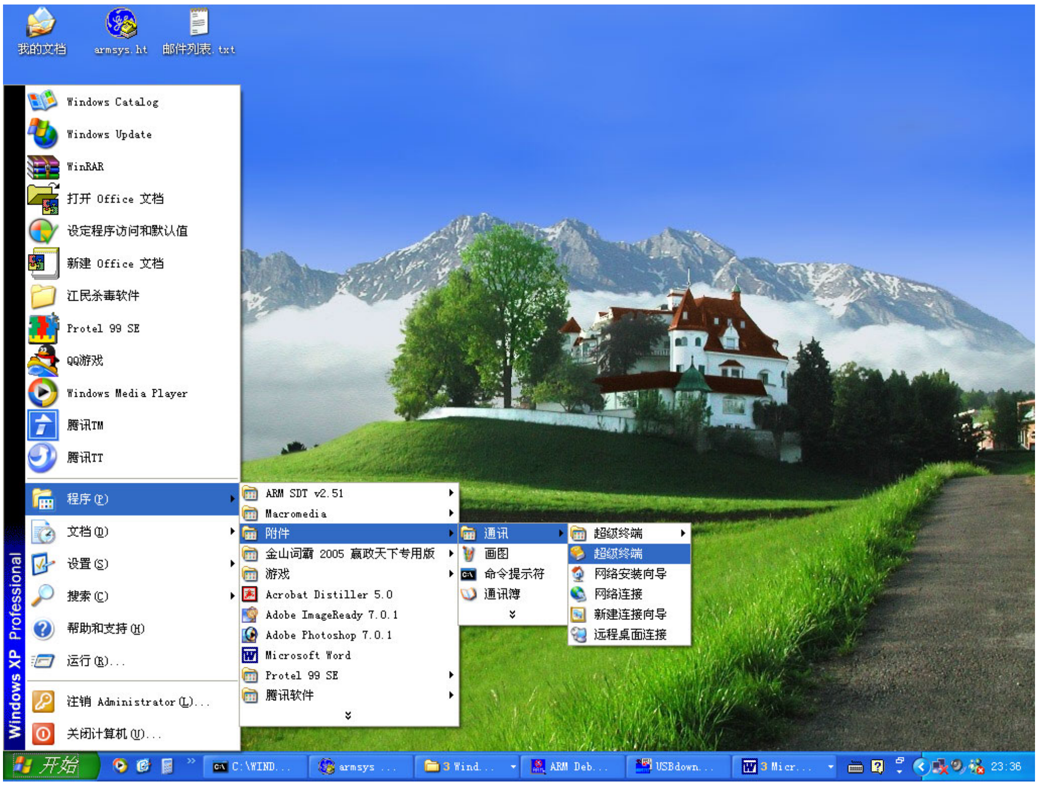


图 3－1 新建超级终端

新建一个超级终端项目，将其命名为 ARMSYS，点击“确定”，弹出以下对话框：



图 3－2 超级终端属性

在“连接时使用”项中选择你所使用的串口号，点击“确定”按钮。按照下图配置该串口：



图 3－3 串口属性设置

点击确定，超级终端就配置好了。最后打开开发板电源即可看到如下启动信息：

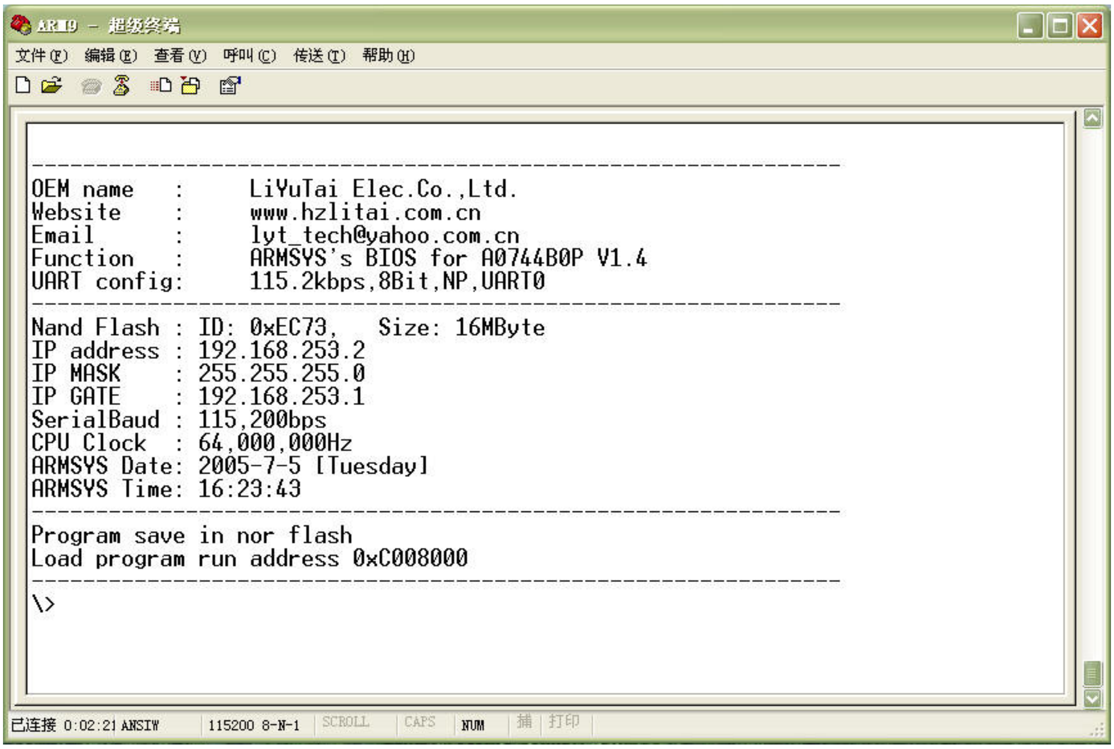


图 3—4 开发板启动信息

3.1.3 开发板接入以太网

用开发板配套的交叉对等网线一端接计算机另一端接开发板，或者用直连网线连接 HUB 集线器到开发板的 RJ45 接口座。打开电源后会看到以太网单元电路中的工作状态指示 LED 会点亮或闪烁，说明开发板已经连接到以太网了。

3.2 BIOS 命令简表

BIOS 具有启动、引导，下载、烧写，设置日期、时间，设置工作频率等多种功能，并且支持引导运行 uCLinux 操作系统。BIOS 启动后随即进入 shell 模式。用户要启动任何功能都必须输入特定的命令行。因此要很好地使用 BIOS,用户必须首先熟悉一些命令。BIOS 支持的命令行及功能说明如下(蓝色为常用功能)：

BIOS 命令	命令行提示信息	功能说明
help	show this list	显示 BIOS 命令列表
?	help	显示 BIOS 命令列表
oem	show corporation information	显示公司有关信息
uclinux	run the uCLinux OS	运行开发板上的 uCLinux 操作系统
test	Run ARMSYS test program	运行开发板测试程序
date	show or set current date	显示或设置开发板时钟的当前日期
time	show or set current time	显示或设置开发板时钟的当前时间

setweek	set weekday	设置星期信息
clock	show system running clock	显示系统当前运行时钟
setmclk	set system running clock	设置系统当前运行时钟
chguart	change uart(0/1)	选择串口（UART0/UART1）
setbaud	set baud rate	设置串口通讯波特率
ipcfg	show or set current IP address	显示或设置当前开发板 IP 地址
netload	download file by net	通过以太网下载文件
netrun	download file by net and run	通过网络下载文件并运行该文件
go	download file by net and run	通过网络下载文件并运行该文件
comload	download file by uart	通过串行口下载文件
comrun	download file by uart and run	通过串行口下载文件并运行该文件
xmload	download file by xmodem	通过 XMODEM 下载文件
xmrun	download file by xmodem and run	通过 XMODEM 下载文件并运行
run	run program	运行程序
move	move data from addr1 to addr2	从 ADDR1 转移数据到 ADDR2
md	show memory data	显示内存数据
senv	save enviroment value to flash	在 NandFlash 存储器保存环境变量
nformat	format NandFlash	对 NandFlash 进行格式化
nfw	write file to NandFlash	写入文件到 NandFlash
nfr	read file from NandFlash to SDRAM	从 NandFlash 中读取文件到 SDRAM 中
nfd	display file from NandFlash	显示出 NandFlash 中的文件
nfdel	delete file from NandFlash	删除 NandFlash 中的文件
defset	default setting	默认设置系统参数

3.3 上位机 FTP 工具

开发板启动 tftp 接收后，要在 PC 端执行 tftp 下载程序，

- 在 windows 2000 或 windows XP 下，直接在“运行”中输入 tftp -i 192.168.253.2 put 文件名（包括路径）即可；
- 在 windows 98 下，使用 CDROM 里所带的 windows 文件夹中的 tftp 程序；
- 在 Linux 下，使用 CDROM 里所带的 linux 文件夹 tftpcmd 程序。注意进行 tftp 传输时要保证 PC 机和开发板处于同一个 IP 网段内。方法请参考 7.5 节。

3.4 BIOS 命令详解

（注：以下命令所带参数中地址和长度都属 16 进制，不必在前面加 0x）

help 和? 罗列出所有命令并给出简单的说明。

date 显示和设置开发板当前日期，只输入 date 命令则显示日期，输入 date 2005-5-1 则设置当前日期为 2005 年 5 月 1 日。

time 显示和设置开发板当前时间，只输入 time 命令则显示时间，输

入 time 12:10:20 则设置当前时间为 12:10:20。

setweek n 可设置星期几，n 从 1 到 7 表示星期一到星期日。

clock 显示开发板当前的工作频率。

setmclk 改变 CPU 工作频率，具体参数设置可见芯片手册，注意不要使频率超出工作范围。

chguart 选择通讯用的串口（UART0/UART1）。

setbaud 改变通讯串口的波特率，改完后要在 PC 上相应改变串口通讯波特率后再敲回车。

ipcfg 可显示和修改 tftp 下载时所用的 IP 地址，只输入 ipcfg 则显示当前 IP 地址，输入 ipcfg 192.168.253.2 则将 ip 地址改为 192.168.253.2。

netload 启动 tftp 接收，若没带地址参数，则使用缺省下载地址 0x0C008000；若指定地址，下载数据保存到指定地址开始的 SDRAM 中去，如 netload c300000。

netrun(go) 启动 tftp 接收完数据后会自动运行下载的程序，缺省下载地址和指定参数同 netload。

comload 启动串口下载（与 PC 机端 DNW 程序配套的串口下载），缺省下载地址和指定参数同 netload。

comrun 启动串口下载（与 PC 机端 DNW 程序配套的串口下载）并在接收完数据后自动运行下载的程序，缺省下载地址和指定参数同 netload 。

xmload 启动 XMODEM 方式下载，可在超级终端内选择 1K XMODEM 或 XMODEM 发送数据到开发板上，缺省下载地址和指定参数同 netload 。

xmrun 在启动 XMODEM 方式接收完数据后自动运行下载到的程序，缺省下载地址和指定参数同 netload 。

run 可运行存储器中的程序，缺省地址就是缺省下载地址，也可指定运行地址。

move 如 move addr1 addr2 size 可将存储器中 addr1 开始的长度为 size 的数据拷贝到 addr2 开始的地址去。

md 显示存储器中的数据，可以带地址参数。

nformat 对开发板上的 NandFlash 进行格式化。

nfw 将下载到 SDRAM 中的文件写入到 NandFlash 中，需先把要写入的文件下载到 SDRAM 中去，写入时要输入文件名字（不能超过 8 个字符），如 nfw test addr size，该命令的意思是将下载到 addr 开始地址中的数据文件，长度为 size，写入到 NandFlash 中去，以文件名 test 保存。

nfr 从 NandFlash 中读取文件到 SDRAM 中，读取的文件要存在于 NandFlash 中，如 nfr test addr r ,该命令是从 NandFlash 中读取文件名为 test 到 SDRAM 的地址为 addr，如果带参数 r 时，读取完文件后直接运行。

nfd 显示出 NandFlash 中的文件。

nfdel 删除 NandFlash 中的文件。如 nfdel test，将 NandFlash 中的文件名为 test 的文件删除。

senv 保存所有参数到 NandFlash 中。

defset 将默认设置一次性设置好。

第四章 ARM 开发环境介绍

4.1 ADS1.2 集成开发环境简介与安装

ADS1.2 是一个使用方便的集成开发环境，全称是 ARM Developer Suite v1.2 。它是由 ARM 公司提供的专门用于 ARM 相关应用开发和调试的综合性软件。在功能和易用性上比较 SDT 都有提高，是一款功能强大又易于使用的开发工具。以下就我们对 ADS1.2 进行一些简要的介绍。

ADS 囊括了一系列的应用，并有相关的文档和实例的支持。使用者可以用它来编写和调试各种基于 ARM 家族 RISC 处理器的应用。你可以用 ADS 来开发、编译、调试采用包括 C、C++和 ARM 汇编语言编写的程序。

ADS 主要由以下部件构成：

◇ 命令行开发工具；

◇ 图形界面开发工具；

◇ 各种辅助工具；

◇ 支持软件。

其中重点介绍一下图形界面开发工具。

◇ **AXD** 提供给基于 Windows 和 UNIX 使用的 ARM 调试器。它提供了一个完全的 Windows 和 UNIX 环境来调试你的 C、C++和汇编语言级的代码。

◇ **Code Warrior IDE** 提供基于 Windows 使用的工程管理工具。它的使用使源码文件的管理和编译工程变得非常方便。但 CodeWarrior IDE 在 UNIX 下不能使用。

运行开发板配套光盘中的开发工具下的 ADS1.2 文件夹下面的可执行安装程序 setup.exe，然后就像安装其他 Windows 应用程序一样一步一步向下执行，最后装好 license.dat 文件，编译调试环境就安装好了。

4.2 JTAG 调试代理软件的安装与使用

4.2.1 首次使用时先安装驱动(以后不用再安装)，将配套光盘中开发工具\ARMJtagDebugFinal 文件夹拷贝到硬盘一个目录下（注意：ARMJtagDebugFinal 所在的路径不要带有中文字符或特殊字符，或者处于太深的目录下），去掉只读属性。然后进入 ARMJtagDebugFinal 目录，双击下的“安装驱动.exe”安装好并口驱动和 OCX。安装过程如图所示：


My E-mail: sendtoyou@21cn.com
InStall Parallel Port Driver
未安装驱动程序
Service is not installed.
Install
Remove
Install OCX
Install
Quit
①
E-mail: sendtoyou@21cn.com


My E-mail: sendtoyou@21cn.com
InStall Parallel Port Driver
观测到的结果
Service is installed and ru
Install
Remove
单击按键
Install OCX
Install
Quit
②
E-mail: sendtoyou@21cn.com



出现以上窗口后就说明 ARM 调试代理软件已经安装成功，点击[确定][Quit]退出即可。

4.2.2　安装完毕后即可运行 ARM7.exe 调试 ARM7 系统。

4.2.3　如果在以后的使用过程中发现程序无法启动，重新执行“安装驱动.exe”即可。

4.2.4　用 JTAG 仿真调试之前，首先将开发板电源接上，JTAG 复位同步跳线帽插上，用开发板配套的并口电缆连接计算机和 JTAG 仿真小板，再用 20 针排线连接 JTAG 仿真小板到开发板的 20 针 JTAG 接口座。

硬件连接好以后，打开开发板电源，运行 ARM7.exe 调试代理软件，可以看到以下结果，说明 JTAG 调试代理软件运行成功。



4.3 使用 CodeWarrior 建立工程并进行编译

首先我们学习如何使用 ADS 中的 CodeWarrior ——项目管理器来管理源代码。一个嵌入式系统项目通常是由多个文件构成的，这其中包括用不同的语言（如汇编或 C）、不同的类型（源文件，或库文件）的文件。CodeWarrior 通过“工程（Project）”来管理一个项目相关的所有文件。因此，在我们正确编译这个项目代码以前，首先要建立“工程”，并加入必要的源文件、库文件等。

4.3.1 调入模板或重新建立项目

我们通常采用工程模板来建立新的工程，工程模板已经针对目标系统对编译选项进行了设置，为避免重复设置，我们提供了一个在 ARMSYS44B0-P 上使用的通用工程模板——template.mcp 。

点击 CodeWarrior 菜单[File | open…]，找到 source\template.mcp（配套光盘中的源程序要拷贝到计算机的硬盘上，并去掉只读属性），选中并打开。

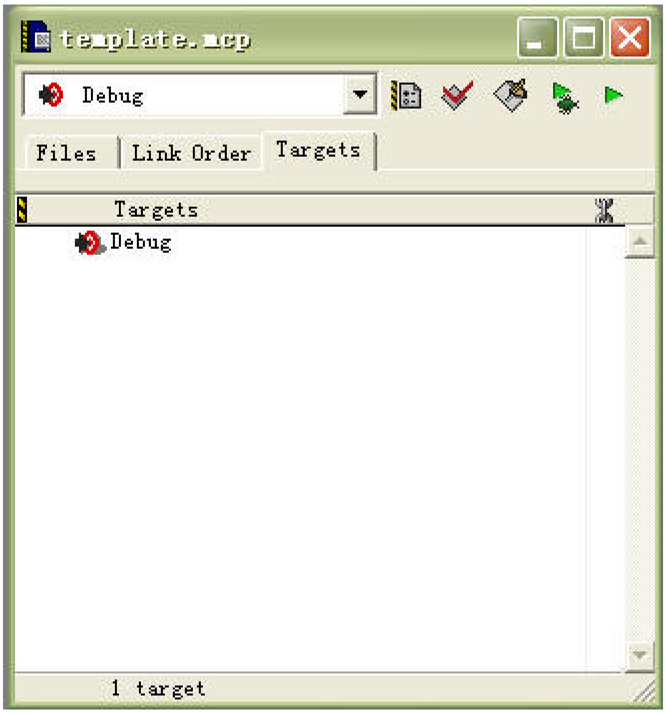


图 4—1 打开工程模板

点击[File | Save as…] ，将它另存为：source\Myhelloworld.mcp， （或者是自定义的其它目录）\ Myhelloworld.mcp 。然后，关闭当前的工程，重新调入 Myhelloworld.mcp ，就可以向工程中添加文件了。

如果你不想利用模板，也可以按照以下步骤来新建一个工程：

(1)选择 File 菜单下的 new 选项，或直接单击，出现以下对话框：

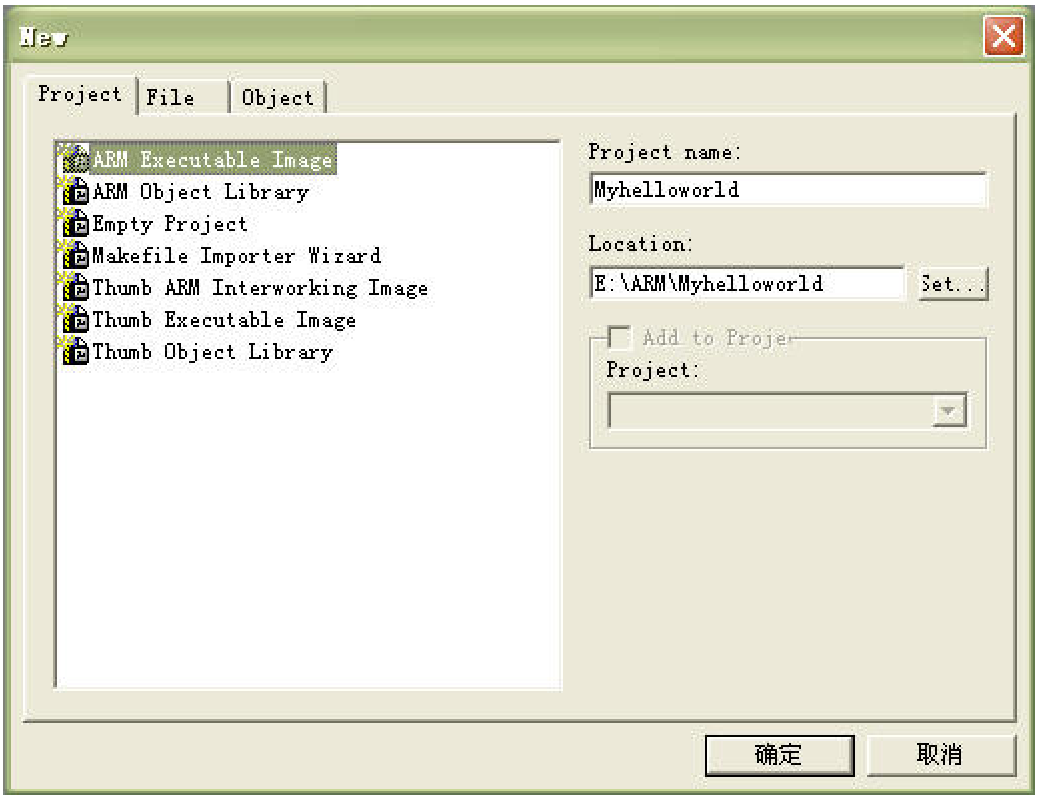


图 4—2 新建工程对话框

（2）选中“ARM Executable Image”选项，在右边的编辑框中输入工程名（例如 Myhelloworld），在下面的 Location 栏中，点击“Set…”，选择放置工程的路径。ADS1.20 不支持中文的目录名字，所以新建工程的文件夹向上一直到根目录的所有文件夹的名字都是英文的。

（3）点击[确定]后工程被建立。

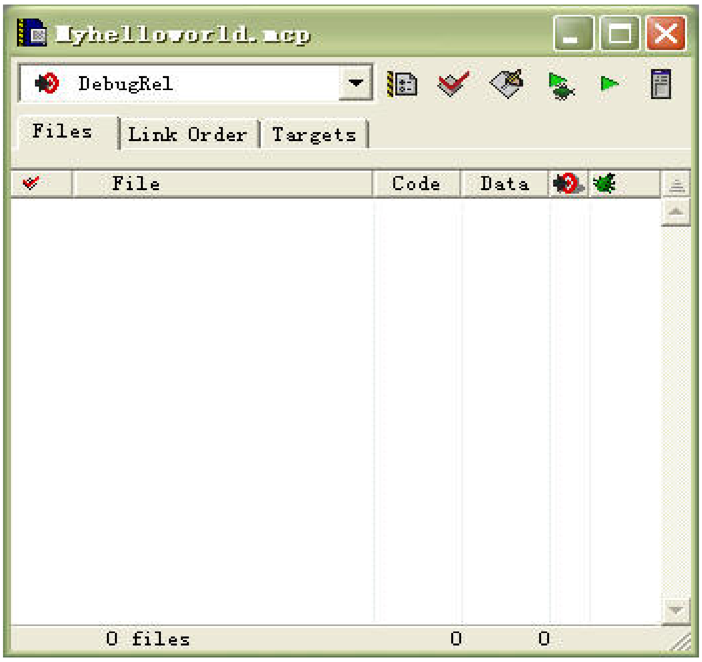


图 4—3 新建工程

但这样的工程还并不能正确地被编译，还需要对工程的编译选项进行适当配置。为了设置方便，先点选 Targets 页面，选中 DebugRel 和 Relese 变量，按

下 Del 键将它们删除，仅留下供调试使用的 Debug 变量。点击菜单[Edit | Debug Setting…]，弹出配置对话框：

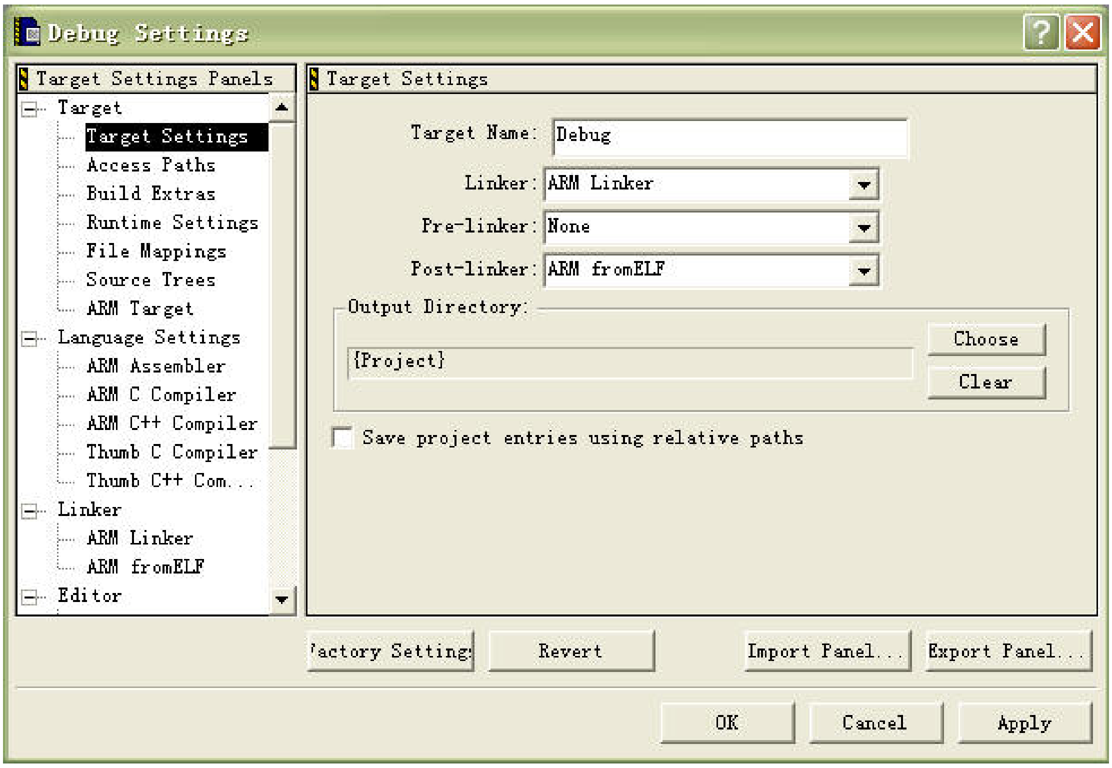


图 4—4 工程配置对话框——目标设置

首先选中 Target Setting ，将其中的 Post-linker 设置为 ARM fromELF ，使得工程在链接后再通过 fromELF 产生二进制代码。

然后选中 ARM Linker，对链接器进行设置：

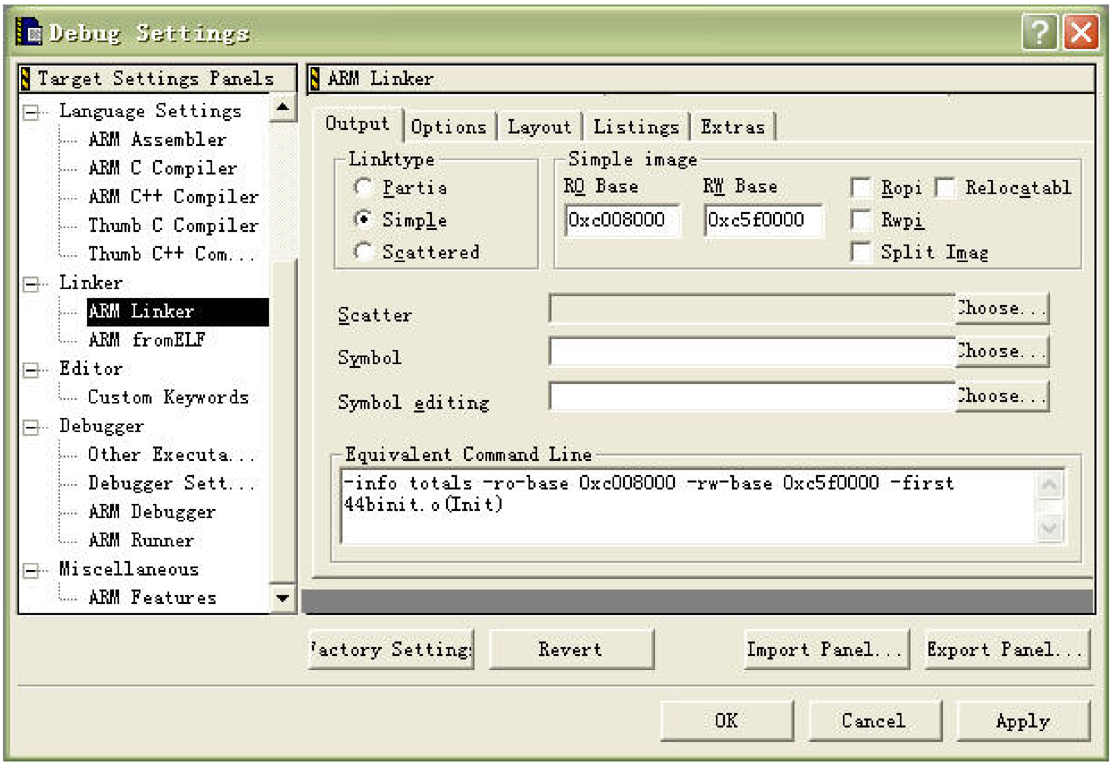


图 4—5（a）ARM Linker 的设置

注意，在调试时，-RO-Base 的设置应当大于 0xC000000 。我们为了与 uCLinux 的 memory map 保持一致，采用了 0xC008000 这个地址。

选取 Layout 页面进行设置：

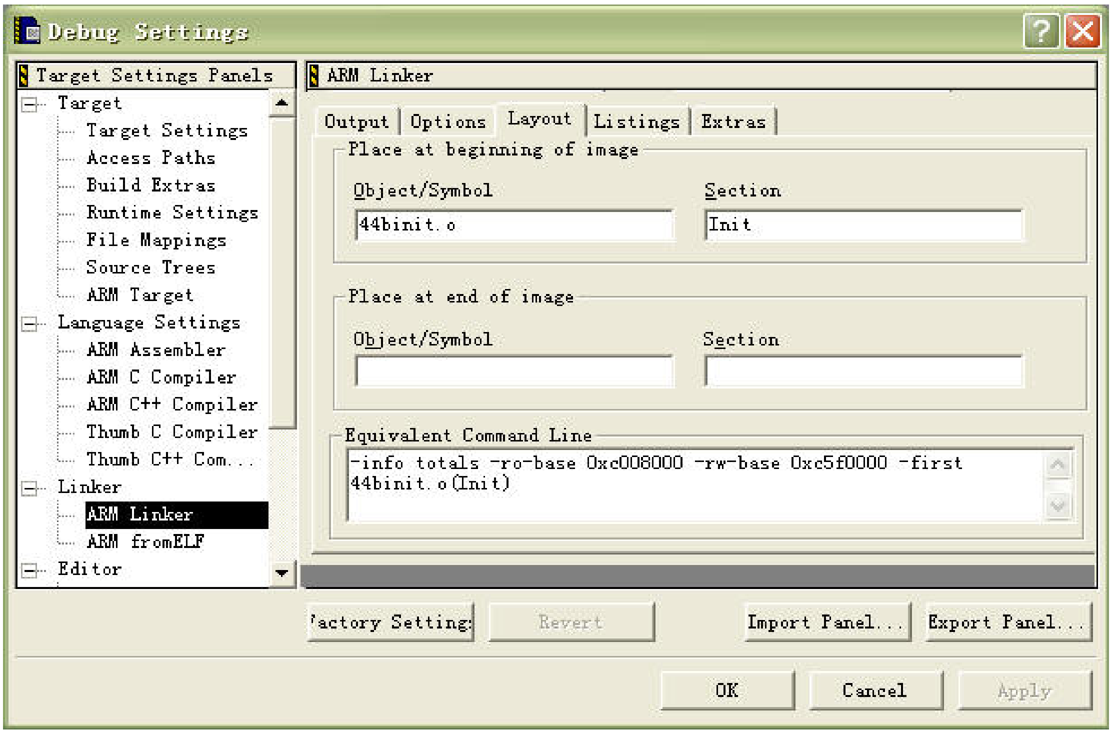


图 4—5（b）ARM Linker 的设置

将 44binit.o 放在映象文件的最前面，它的区域名是 Init。

最后，如果你希望编译的最后生成二进制文件，就要设置 ARM fromELF：

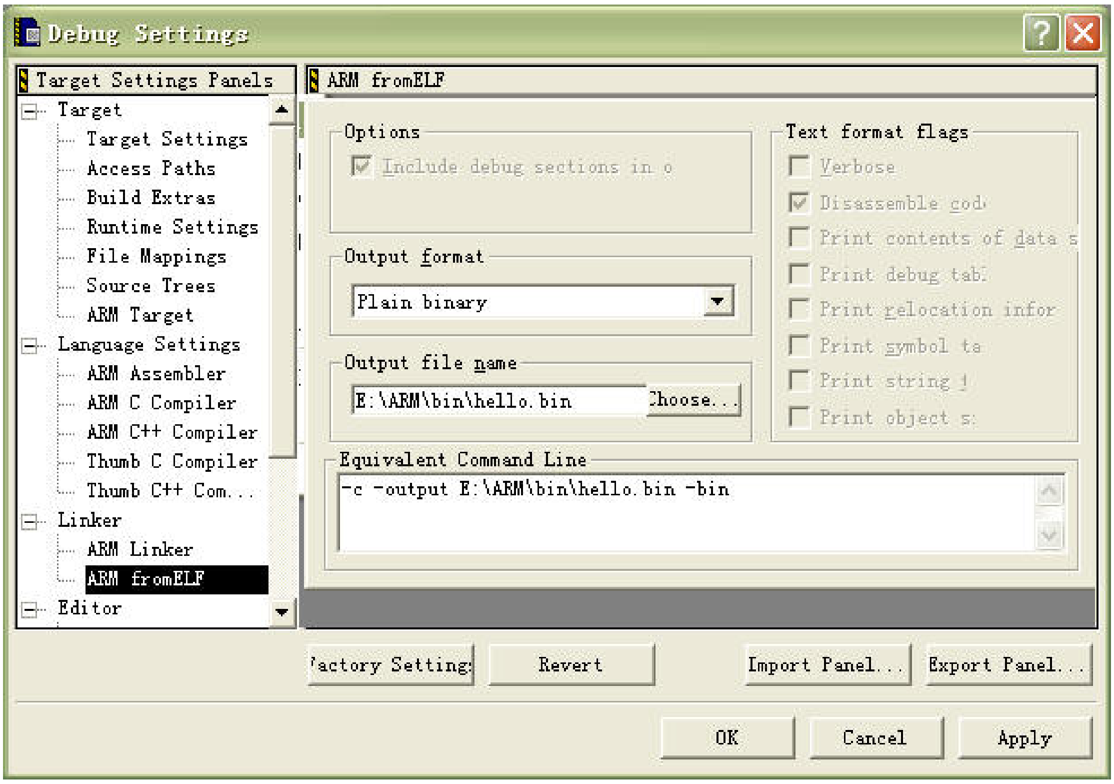


图 4—6 ARM fromELF 的设置

在 Output format 栏中选择 Plain binary ，在 Output file name 栏中，点击“Choose… ”选择你要输出的二进制文件的文件名和路径。

这样，对于 Debug 变量的基本设置都完成了。点击“OK”键退出。

4.3.2 在工程中添加源文件

在图 4-2 的对话框中，点选 File 页面，选中 Text File，并设置好文件名和路径，点击确定，CodeWarrior 就会为你新建一个源文件，并可以开始编辑该空文件。CodeWarrior 与 SDT 中的 APM 不同，它具有一个很不错的源代码编辑器，因此，大多数时候，我们可以直接采用它的代码编辑器来编写好程序，然后再添加到工程中。

添加源文件的步骤如下:例如添加 main.c 文件,在图 4—3 窗口中点选 Files 页面，在空白处单击鼠标右键，点选“Add Files”项，从目录中选取 main.c 文件（Myhelloworld\main.c），点击“打开”，main.c 文件就被加入了工程中。

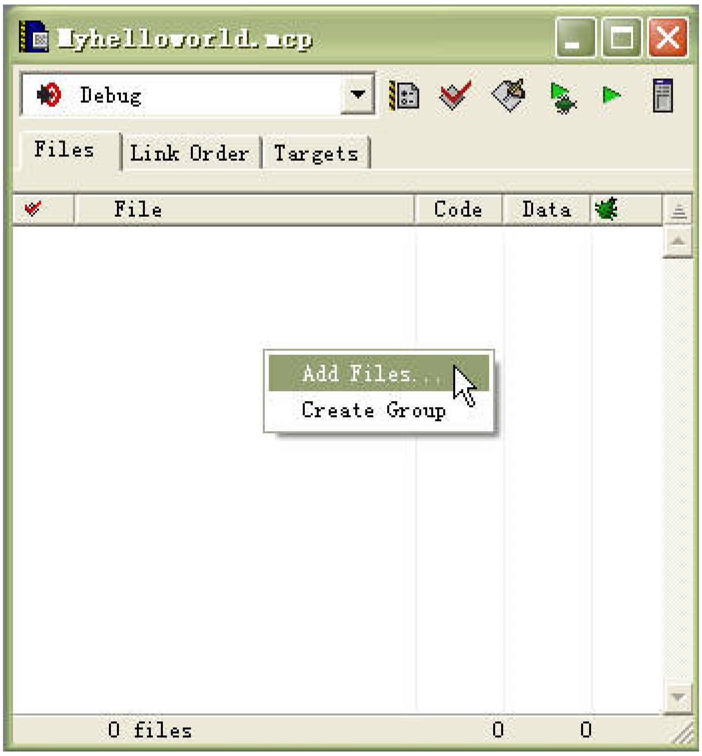


图 4—7 添加源文件

用同样的方法，将 Myhelloworld \下所有的*.C 和*.S 源文件文件都添加到 source 中去(包括 Target 目录下的源文件)。

Target 目录下还有一个 44blib.c 文件，这是一个库文件，其中提供了一些常用函数的定义，这些函数在 44blib.h 进行了声明。这个文件也必须添加到工程中。同样的方法，按鼠标右键，Add files …，将 44blib.c 文件添加到工程中。所有必须的文件添加完成后如图 4—8 所示。

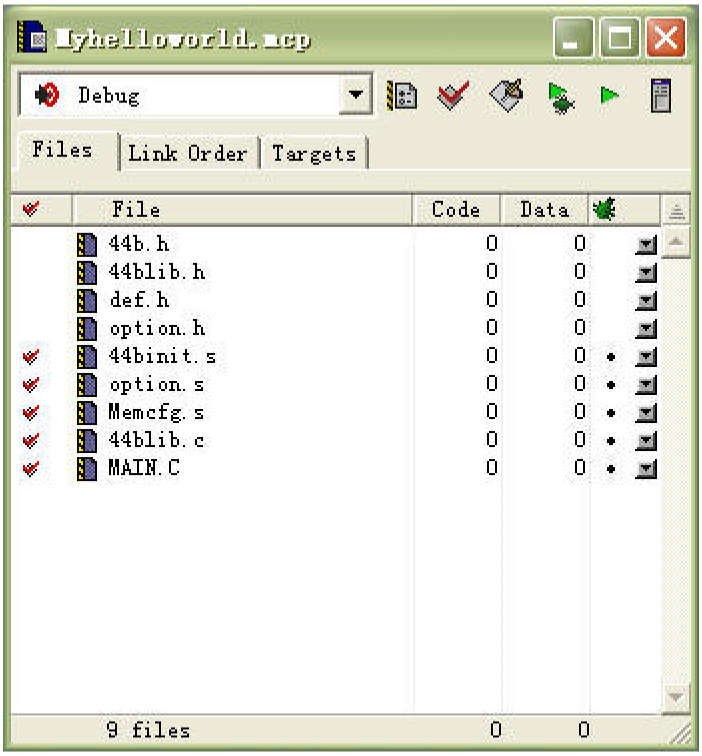
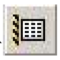


图 4—8 源文件添加完成

4.3.3 工程进行编译和连接

注意到在上图 4—8 中新加入的文件前面有个红色的“钩”，说明这个文件还没有被编译过。在进行编译之前，你必须正确设置该工程的工具配置选项。如果前面采用的是直接调入工程模板，有些选项已经在模板中保存了下来，可以不再进行设置。如果是新建工程，则必须按照上文中所述的步骤进行设置。

- □ 选中所有的文件，点击图标进行文件数据同步；
- □ 然后点击图标，对文件进行编译（compile）；
- □ 点击按钮，对工程进行 Make，Make 的行为包括以下过程：
  - ○ 编译和汇编源程序文件，产生*.o 对象文件；
  - ○ 链接对象文件和库产生可执行映像文件；
  - ○ 产生二进制代码。

Make 之后将弹出“Errors & Warnings ”对话框，来报告出错和警告情况。编译成功后的显示如下。注意到左上脚标示的错误和警告数目都是“0”，如图所示：

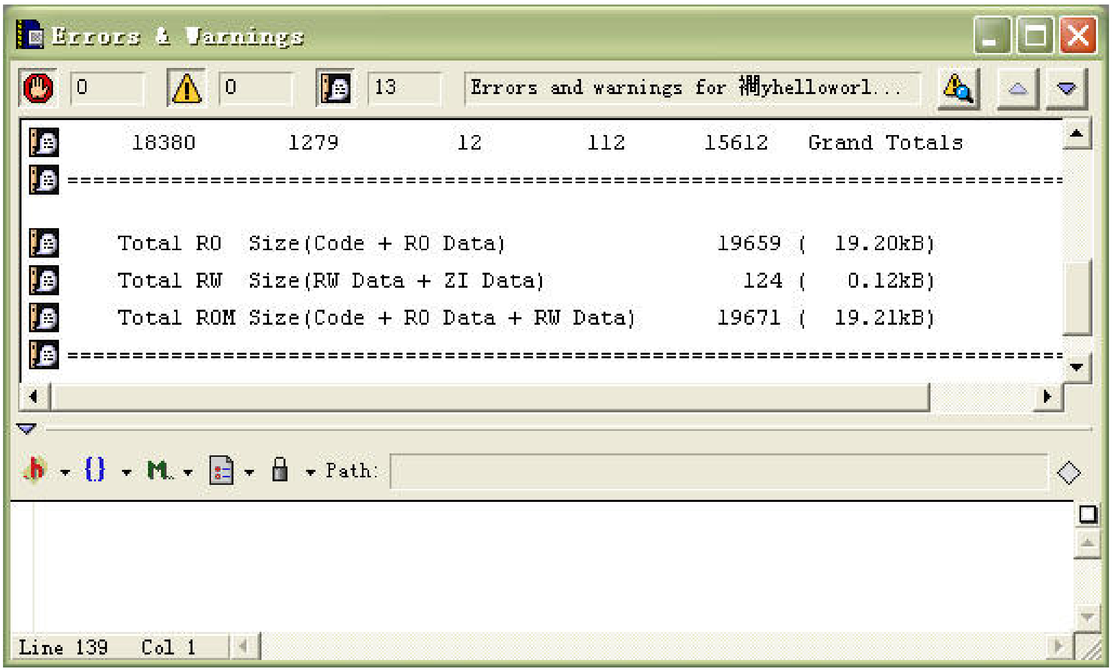


图 4—9 编译后的结果

Make 结束后产生了可执行映象文件 Myhelloworld.axf 文件，这个文件可以载入 AXD 进行仿真调试了。并且还通过 fromelf 工具将 ELF 文件转换为二进制格式文件 hello.bin 。它可以用来最终固化到 flash ROM 中（但链接选项中的-ro-base 要修改为 0x0），也可以下载到-ro-base 地址中运行。

4.4 使用 AXD 进行仿真调试

4.4.1 调试前的准备

在调试之前，我们先用并口电缆将计算机并口和 JTAG 仿真小板连接起来，并把 JTAG 仿真小板和开发板的 JTAG 接口用 20 芯排线连接，用串口线将计算机串口和主板的 UART0 口连接起来。然后检查 JTAG 同步复位跳线帽是否插上，所有连线连接正确、接触良好后就可以拨动电源开关上电了。

在打开电源之前，要先打开按上文介绍建立的“超级终端”才能观察到开发板的启动输出信息。如图 3—4 中所示。

电源打开之后，可以听到主板发出一声蜂鸣器的“嘀——”声，看到绿色发光管点亮后熄灭，有一个绿色发光管在周期闪烁，这说明开发板启动正常。此时 JTAG 仿真小板上只有红色电源指示发光管点亮，说明并口已经连接好了。

在进行调试之前，要先建立好 AXD 与目标系统之间的通讯。如果采用简易 JTAG 调试器进行调试，则首先要运行 JTAG 调试代理软件。注意，在 AXD 调试器在线仿真期间，不要关闭 JTAG 调试代理软件！如果采用 Multi-ICE 仿真器来调试，则首先要运行 Multi-ICE Server（具体请查看仿真器的使用说明）。

4.4.2 AXD 调试器的设置

在 CodeWarrior 编译环境中，工程经过编译成功，产生了*.axf 文件之后，就可以进行调试了。点击 按钮，进入了 AXD 视窗界面。点击菜单项[Option | Configure Target…] ，对调试目标进行配置：

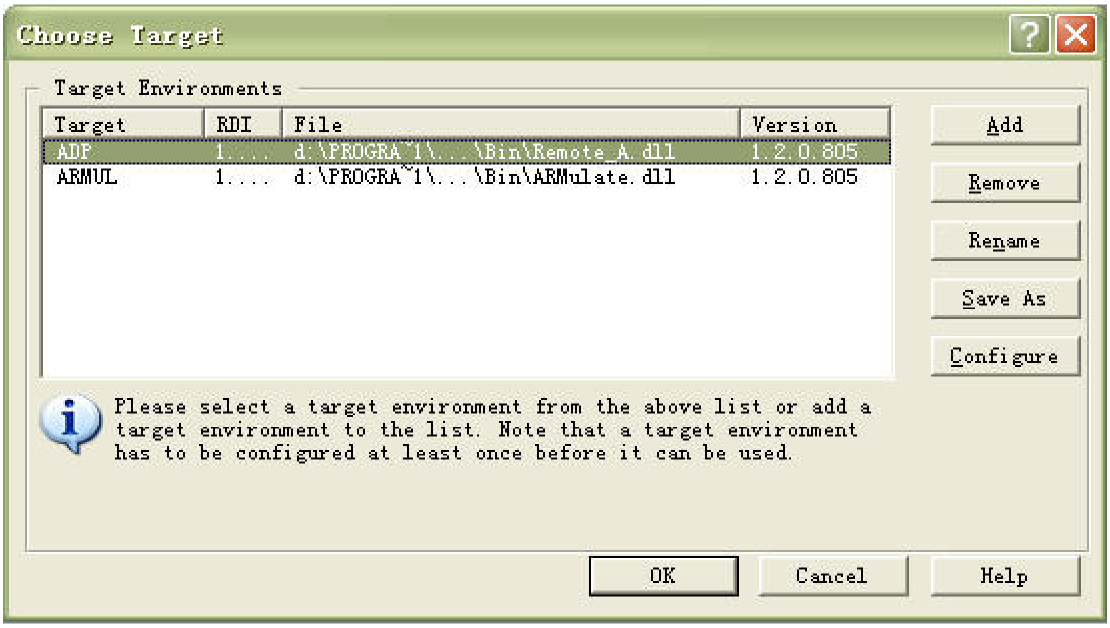


图 4—10 调试目标设置对话框

在 Target Environment 栏中选中“ADP ”选项，注意到下面的注释的说明，“ADP 是直接连接 ARM Debugger 到目标板或者到目标板上的 EmbeddedICE 单元的一种方式。直接连接目标板需要 Angel 调试监视器软件的支持。参考 RDI 1.51”， 点击“Configure”按钮，进入到下面设置：

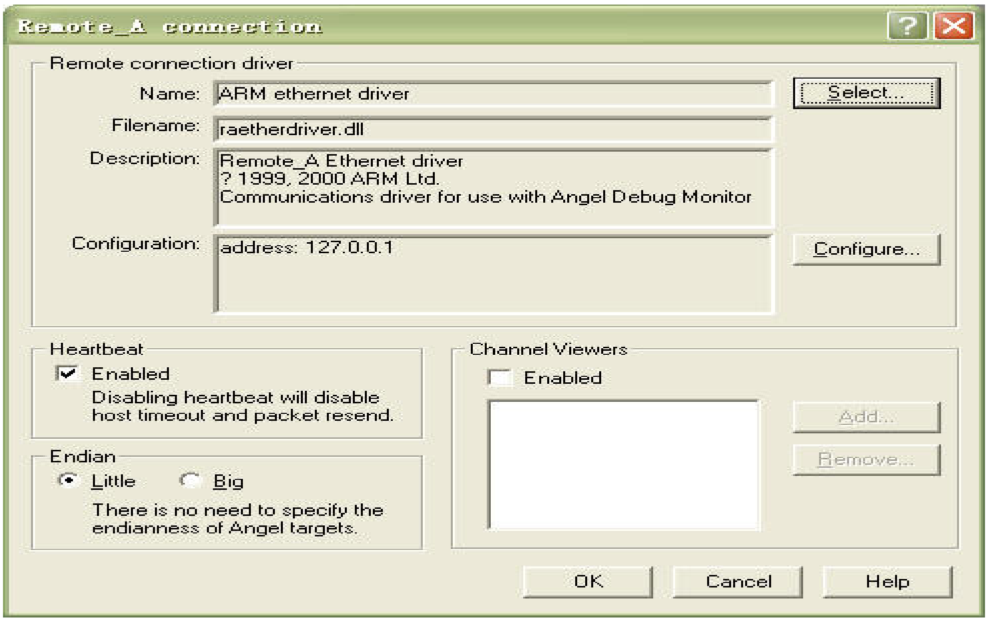


图 4—11 Remote_A 连接设置

其中“Remote Connection driver”栏中，点击右边的“Select…”按钮，

选择“ARM ethernet driver”。点击右边的“Configure… ”按钮，在编辑栏中输入本机的 IP 地址或者 127.0.0.1 。其它设置如上图所示，保持不变。点击“OK”退出调试目标的设置。这时会弹出：



图 4—12 重新载入对话框

点击“是”按钮，如果目标系统正确链接了，会看到程序下载的进度条显示。进度消息框消失后，显示当前执行代码视窗，蓝色指针指向第一条执行的语句：

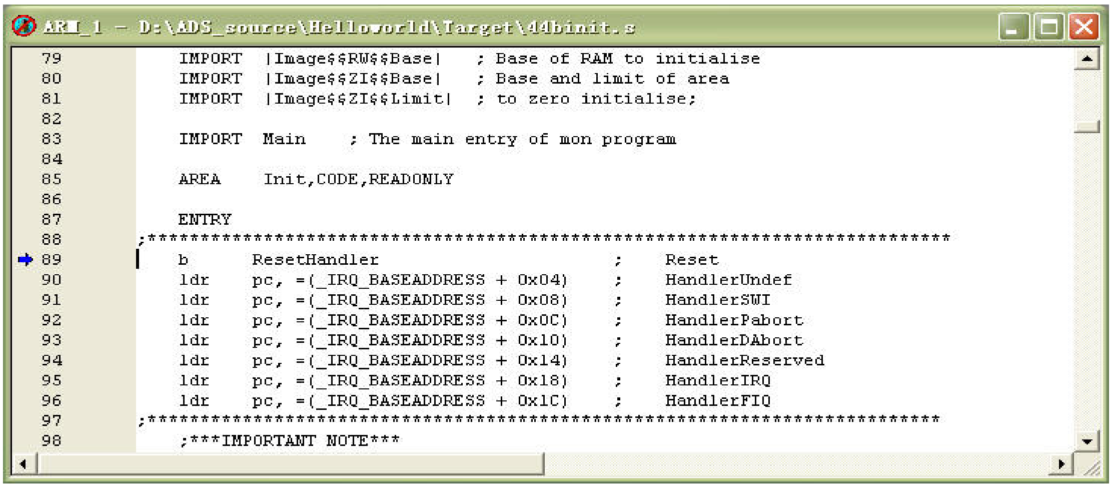
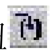


图 4—13 当前执行代码窗口

这时，先点击 按钮，尝试进行单步运行，如果程序立即正确地跳转到“ResetHandler”处执行，而没有跑飞或顺序执行，则说明程序的下载成功了，可以进行调试了。

4.4.3 AXD 调试器的使用

我们首先来熟悉一下断点的设置。下拉滚动条至 377 行，在 BL Main 语句处点击按钮 设置一个断点，如图 4-14。然后点击按钮 （GO），令程序自动执行到断点。当程序执行到 BL Main 语句处，自动停止，点击按钮 ，程序跳转到 main.c 文件的的 Main() 处程序开始运行，如图 4-15。

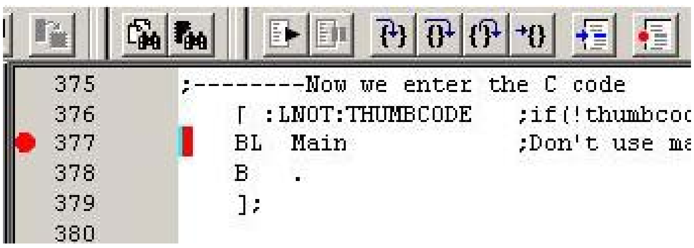


图 4—14 放置断点

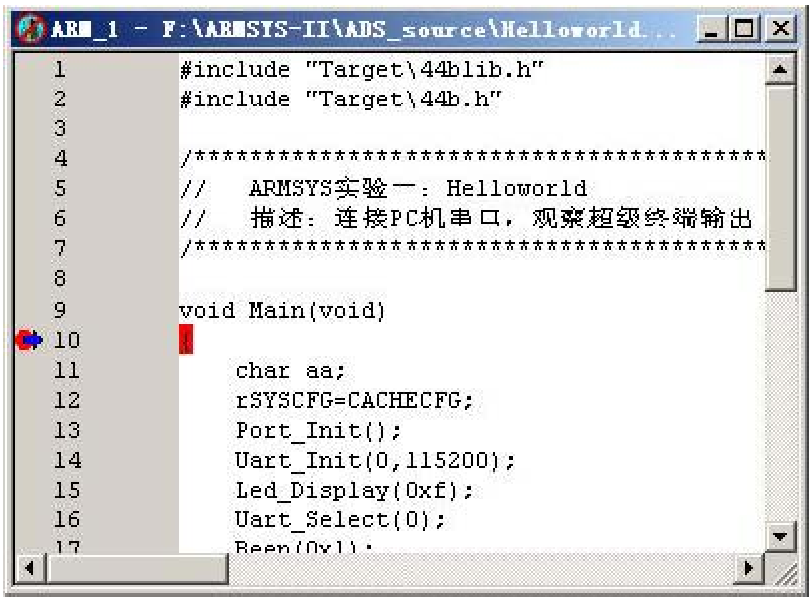


图 4—15 进入主函数运行

通过上面的操作，我们了解到，44binit.s 程序中的 BL Main 语句就是跳转到 C 语言 main（）函数的入口语句。AXD 也会自动在 Main()函数的入口处放置一个断点，因此程序下载后，立即全速运行的话，就会首先跳到该断点停下来。

读者可以继续进行一些单步操作，了解每条语句的作用。

4.4.4 AXD 观测窗口

AXD 提供了许多有用的观察窗口，点击菜单项中的 Processor View ，可以从它的下拉菜单项中了解可观察的项目。

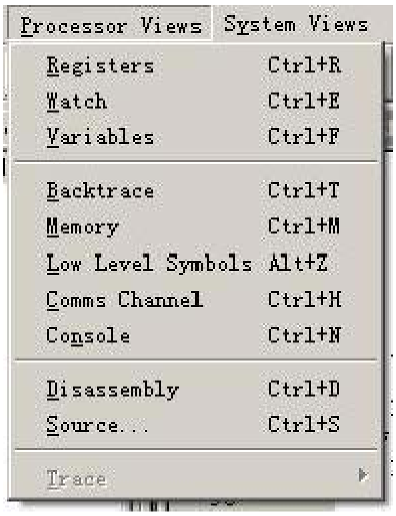


图 4-16 观测窗口菜单

这里说明一下其中常用的项目：

Registers：可以查看 CPU 在各个工作模式下内部寄存器的值；

Variables ：查看变量，本地变量、全局变量、类变量；

Watch ：可以用表达式查看变量的值；

Backtrace：函数调用情况（堆栈）查看；

Memory：查看存储器内容。输入地址，即可查看这个地址开始的存储单元的值。

4.4.5 程序全速运行

在 AXD 中点击 ‘GO’图标，可以全速地运行程序，注意观察超级终端窗体，上面将显示如下信息。

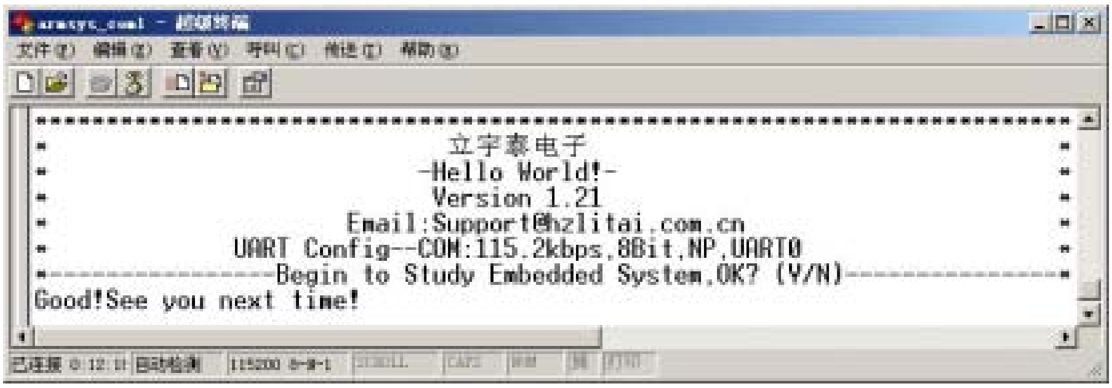


图 4－17 Hello World 运行后超级终端的显示

在超级终端上显示“Begin to Study Embedded System, OK? (Y/N) ”后，在计算机键盘上键入 Y，超级终端上出现“Good! See you next time!”。

第五章 程序代码下载与存储

5.1 程序文件下载至开发板的途径

5.1.1 通过网络下载程序文件

首先将开发板的电源、串口和网线连接好，然后打开电源开关，开发板 BIOS 启动完成以后，在超级终端中输入 BIOS 命令“netload”，如下图所示：

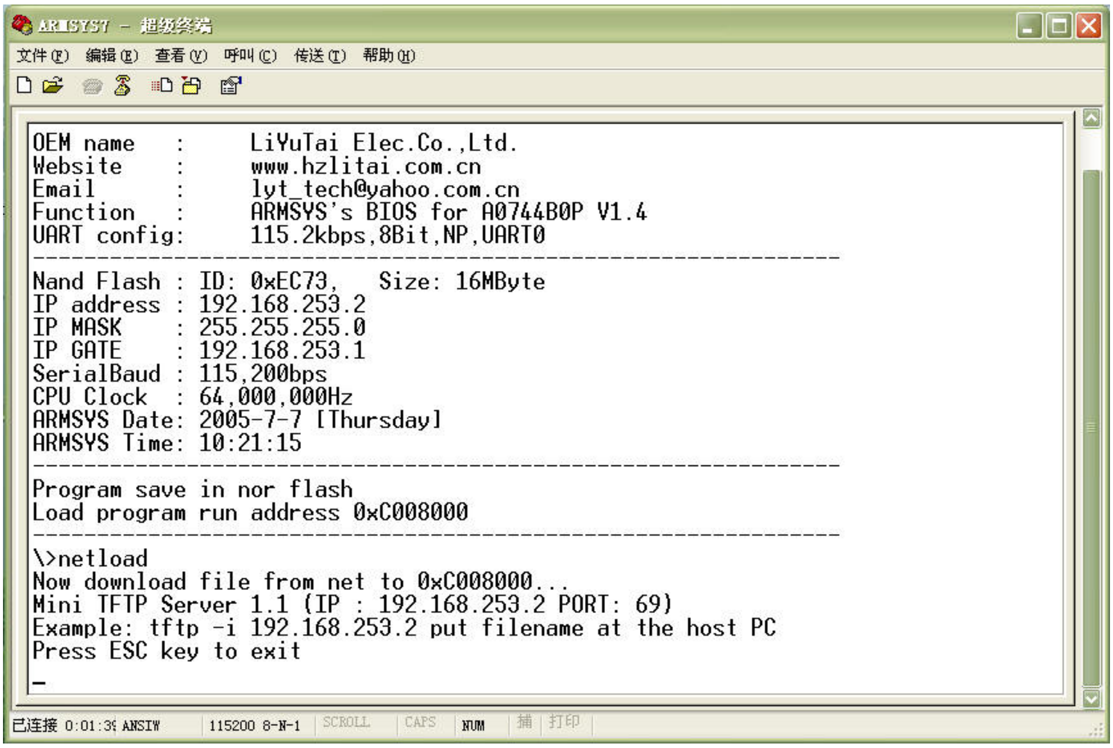


图 5－1 网络下载命令

“netload”命令是启动开发板网络下载程序，可以在命令行中加入下载目的地址，没有输入时默认下载地址为“0xC008000”，如要将程序下载到目的地址为“0xC200000”时，输入命令“netload c200000”即可。另外通过网络下载时，开发板的 IP 地址要和计算机的 IP 地址在同一个 IP 段内。如果不在同一个 IP 段内可以通过命令“ipcfg”更改开发板的 IP 地址，或者改变计算机的 IP 地址。

在出现图 5－1 后，可以在计算机中输入命令或执行批处理文件来下载程序文件了。在计算机中输入的命令行如图下所示：

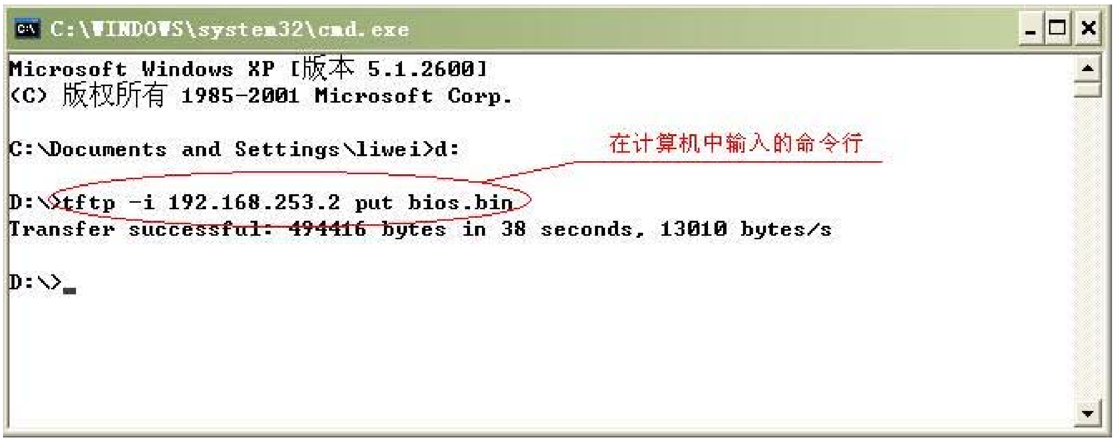


图 5－2 计算机执行下载命令

在计算机中输入完命令回车后，会在超级终端上看到下载的进程，同时开发板的网络指示灯也会快速闪烁，至到程序文件数据传输完成。程序文件传输完成后在计算机的 DOS 窗口中可以看到传输的字节数、总用时和传输速率的信息，而在超级终端中可以看到开发板输出的传输信息，如图所示：

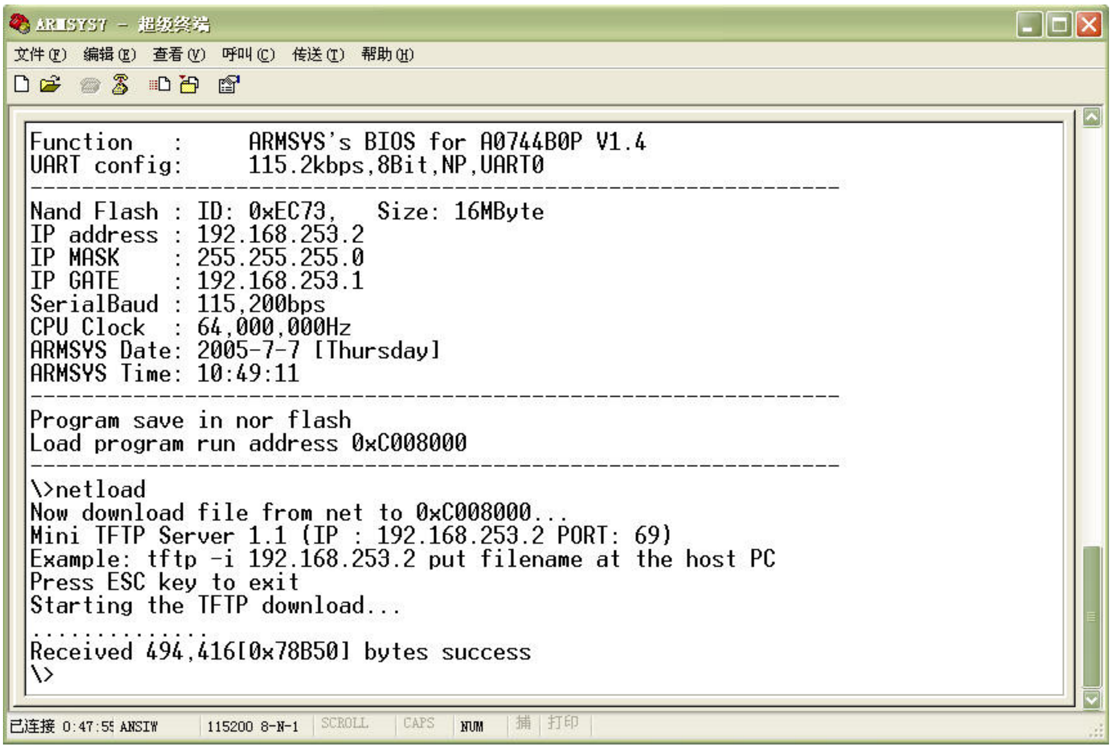


图 5－3 网络下载后的结果

5.1.2 通过串口下载程序文件

将开发板接上电源和串口电缆到计算机，打开计算机上的超级终端和开发板电源，在超级终端中输入命令“xmload”，下载目的地址采用默认的地址“0xC008000”，也可改变下载的目的地址，如“xmload c200000”。如下图所示：

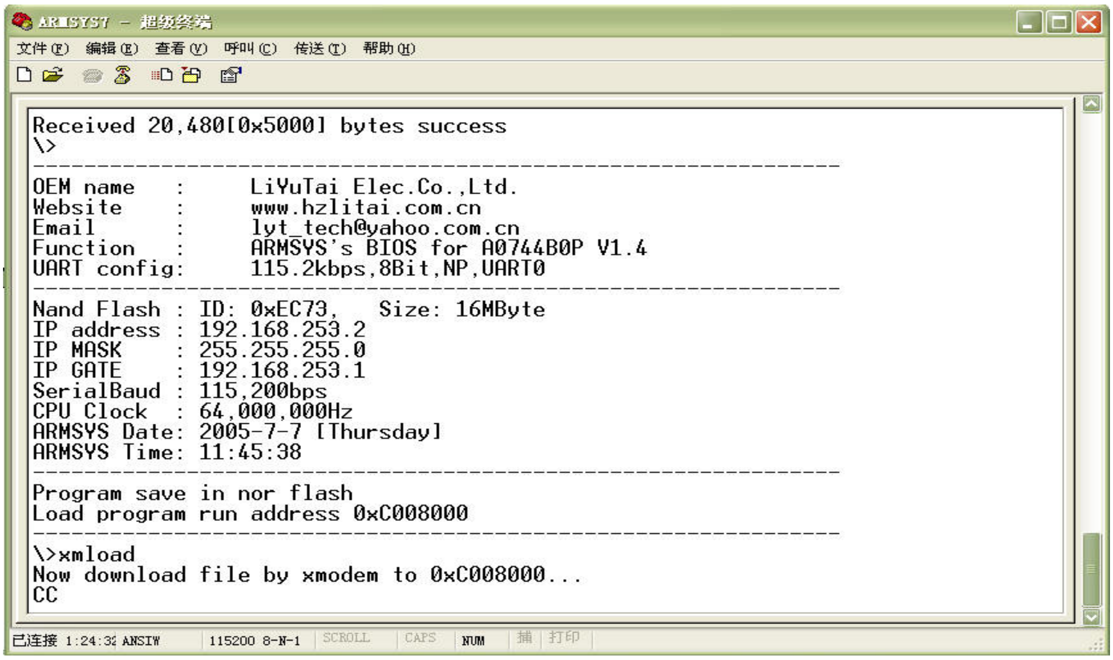


图 5－4 串口 Xmodem 传输程序文件

启动开发板串口 Xmodem 传输程序后，等待计算机运行 Xmodem 程序下载程序文件到开发板。然后在超级终端窗口点击[传送|发送文件]，出现如下窗口：



图 5－5 Xmodem 发送文件

点击[浏览]选择要下载的程序文件，协议选择“Xmodem”或“1K Xmodem”然后点击[发送]即可开始传输程序文件了。可以看到如下图所示的传输过程：

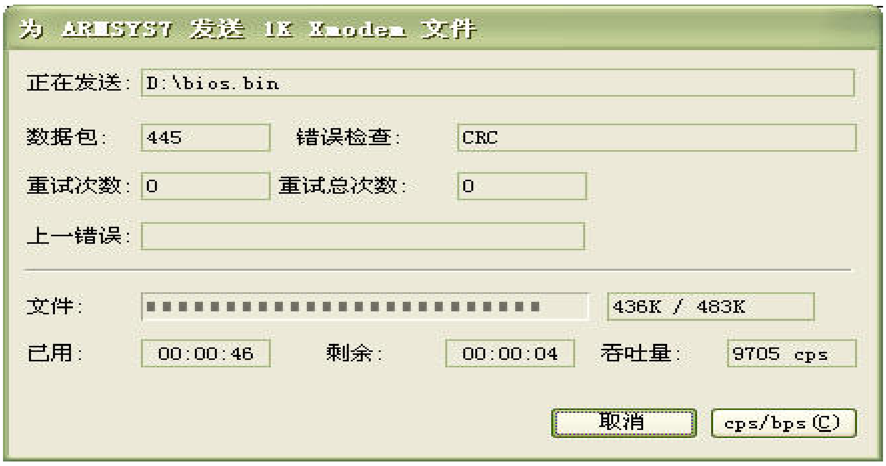


图 5－6 串口 Xmodem 传输过程

当程序文件传输完了，该窗口关闭，通过串口传输完成。

5.2 采用 BIOS 烧写程序代码到 NorFlash 中

ARMSYS44B0-P 开发板的 Bank0 接至 NorFlash 芯片，在开发板启动时就从 NorFlash 的零地址开始度取程序执行。因此，NorFlash 的从零地址开始要存放开发板的启动程序代码。采用 BIOS 烧写程序代码到 NorFlash 时要用到 44bapp.bin 烧写工具，先把该烧写工具下载到开发板的 SDRAM 中，再把被烧写的程序代码下载到地址为“0xC200000”中，代码大小不能超出 NorFlash 的容量，具体步骤如下（以下为系统出厂时的烧录步骤）：

（1）利用网口或串口先将 44bapp.bin 程序代码文件下载到目的地址为“0xC008000”处，等到下面步骤执行完后运行该代码。

（2）下载 ARMSYS44B0_P_BIOS.bin 代码文件到“0xC200000”地址中。

（3）下载 ARMSYS44B0_P_TEST.bin 代码文件到“0xC210000”地址中。

（4）下载 imagerom.bin 代码文件到“0xC230000”地址中。

（5）在超级终端中输入命令“run c008000”回车，开始执行烧写。

烧写完成后，绿色的发光管会全亮。然后复位即执行刚才烧写的程序代码。

5.3 写入程序文件到 NandFlash 中

ARMSYS44B0-P 开发板集成有 NandFlash 芯片，可以存放更多更大的程序文件，存放时只要给文件给定一文件名（不超过 8 个字符）即可写入。要把某个程序文件写入到 NandFlash 中，首先要把它下载到 SDRAM 中去，然后再执行写入命令，这样就把程序代码保存到 Nandflash 上了。掉电不会丢失，随时可以读出到 SDRAM 中执行。

5.3.1 NandFlash 中程序文件的写入

（1）通过网口或者串口下载程序到 SDRAM 中，下载的地址可以自定义。

（2）在超级终端中输入命令“nfw hello”回车，开始把刚下载的程序写入到 NandFlash 中去，直到程序文件写入结束。

注意，写入命令“nfw”除了必须输入要存放的文件名外，还可以加上被写入程序存放在 SDRAM 中的起始地址和字节数（用 16 进制表示）。如命令“nfw hello c200000 5000”是把存放在 SDRAM 从 0xC200000 到 0xC205000 地址中的数据以“hello”为文件名写入到 NandFlash 中。如果不输入起始地址和字节数时，BIOS 取刚才下载的程序文件时使用的下载地址和文件长度值，如果之前没有下载过程序文件，且此次的输入又没有给定起始地址和字节数，则不能写入。

5.3.2 NandFlash 中程序文件的读取与删除

（1）从 NandFlash 中读取程序文件用“nfr”命令，该命令也必须输入要读取的文件名，读取后默认存放地址为“0xC008000”，还可以加入“r”运行读取的程序代码。完整的命令行为“nfr hello c008000 r”，意思是把存放在 NandFlash

中以文件名为“hello”的程序代码读取到 SDRAM 中地址为“0xC008000”处，读取完成后运行该程序代码。

（2）从 NandFlash 中删除已经写入的程序文件用命令“nfdel”再加上要删除的文件名即可。如“nfdel hello”。

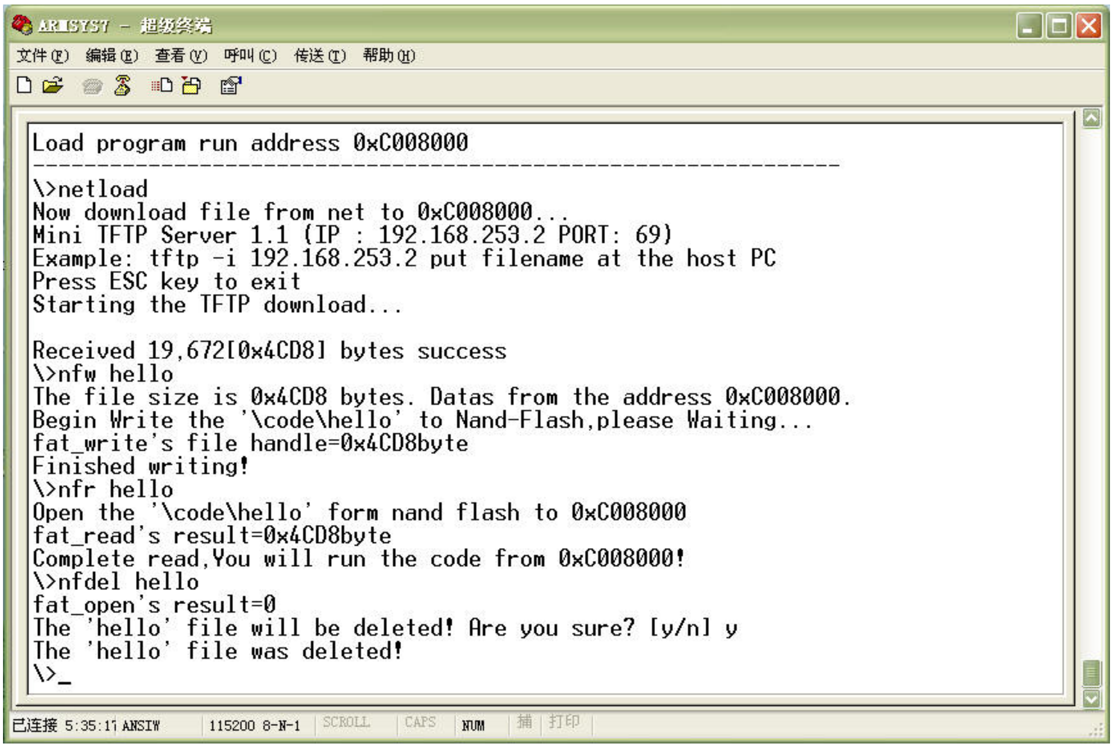


图 6－7 NandFlash 的操作过程

5.4 启动代码的烧写（系统恢复）

5.4.1 使用 FlashPGM 快速烧写 Flash

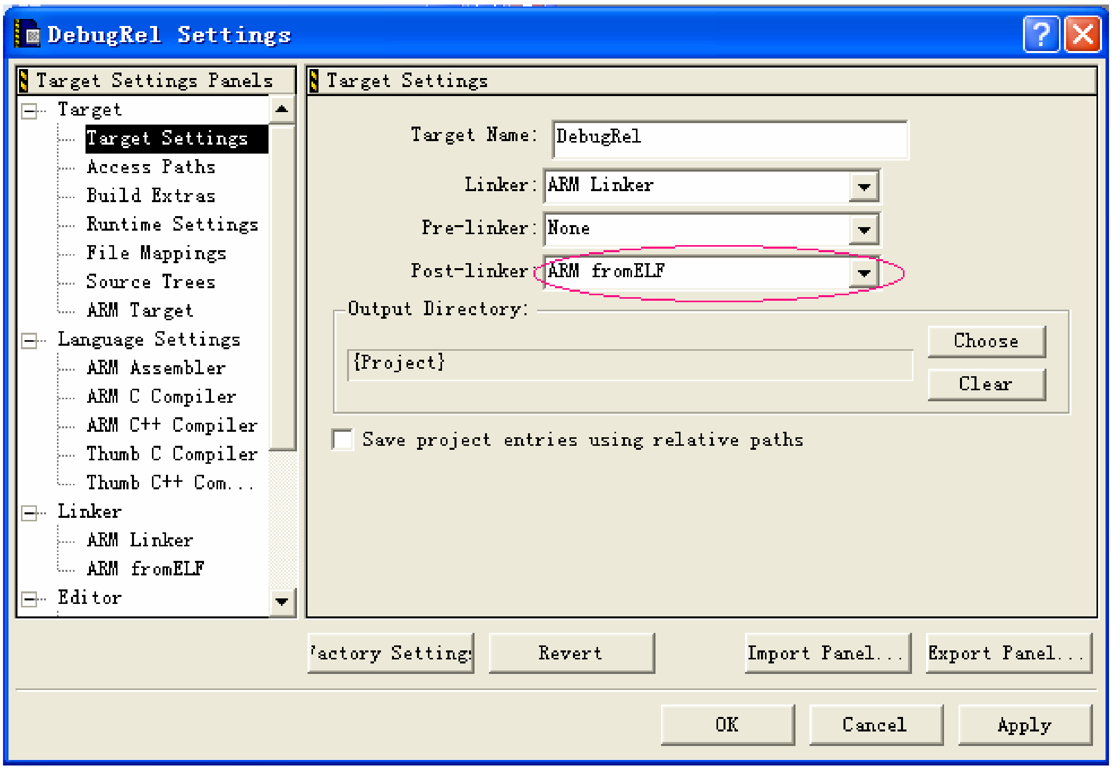
5.4.1.1 计算机的设置

请在 PC 机的 BIOS 里面将并口设置成 EPP 方式；

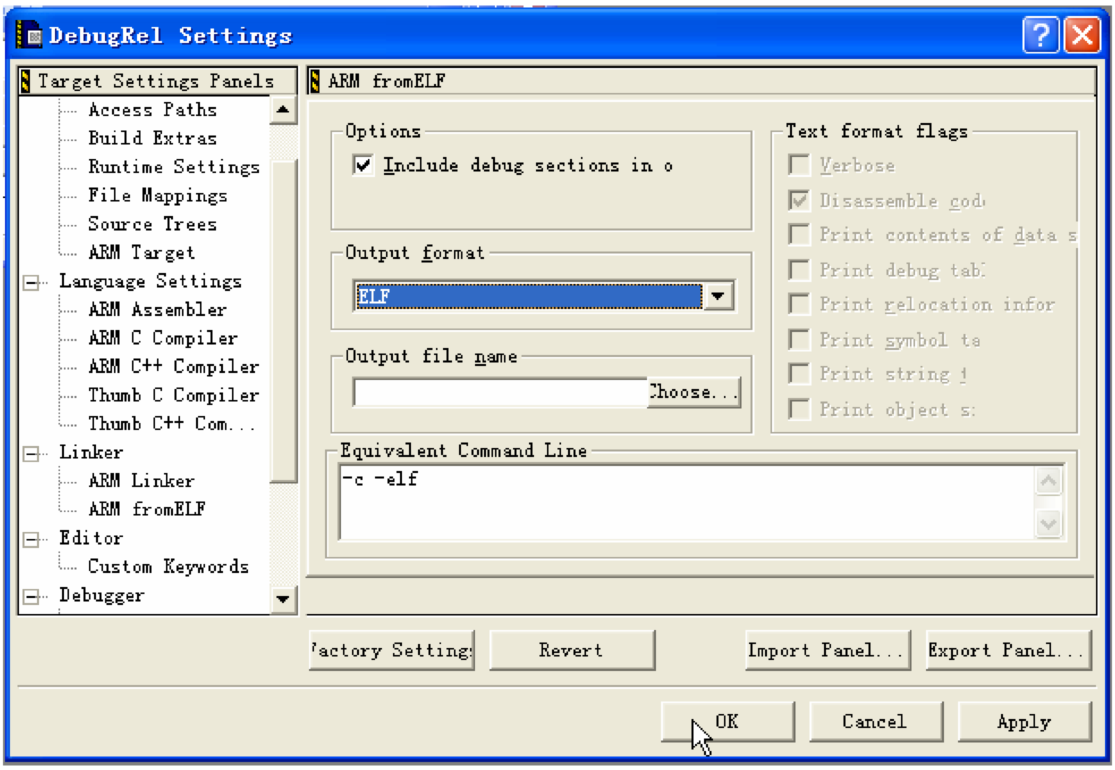
hex 文件的准备

flashpgm 不能烧写*.bin 格式的文件，但可以烧写*.elf 或者是*.hex 格式的文件，一般都使用*.elf 格式的文件进行烧写。

以 ADS1.2 为例，在其项目编译界面下按 ALT＋F7 进入下面的设置界面，按照下面画红色圆圈的画面在 Target Settings 目录下选择 ARM fromELF ，以生成目标代码：



然后在 Linker 目录下选择 ELF。

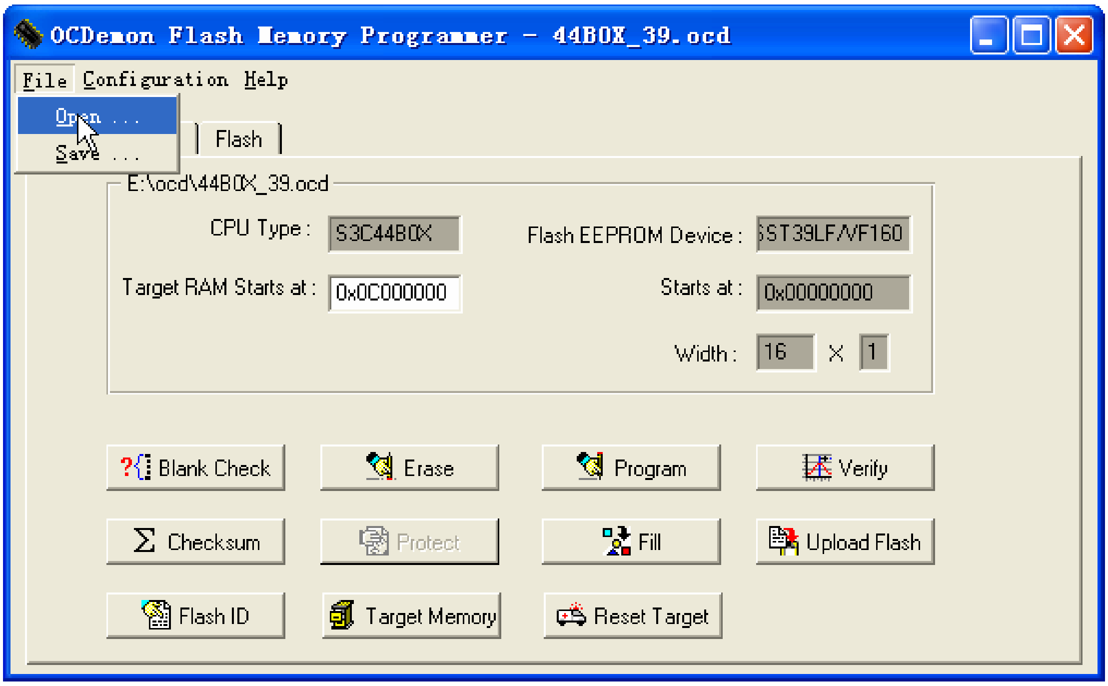


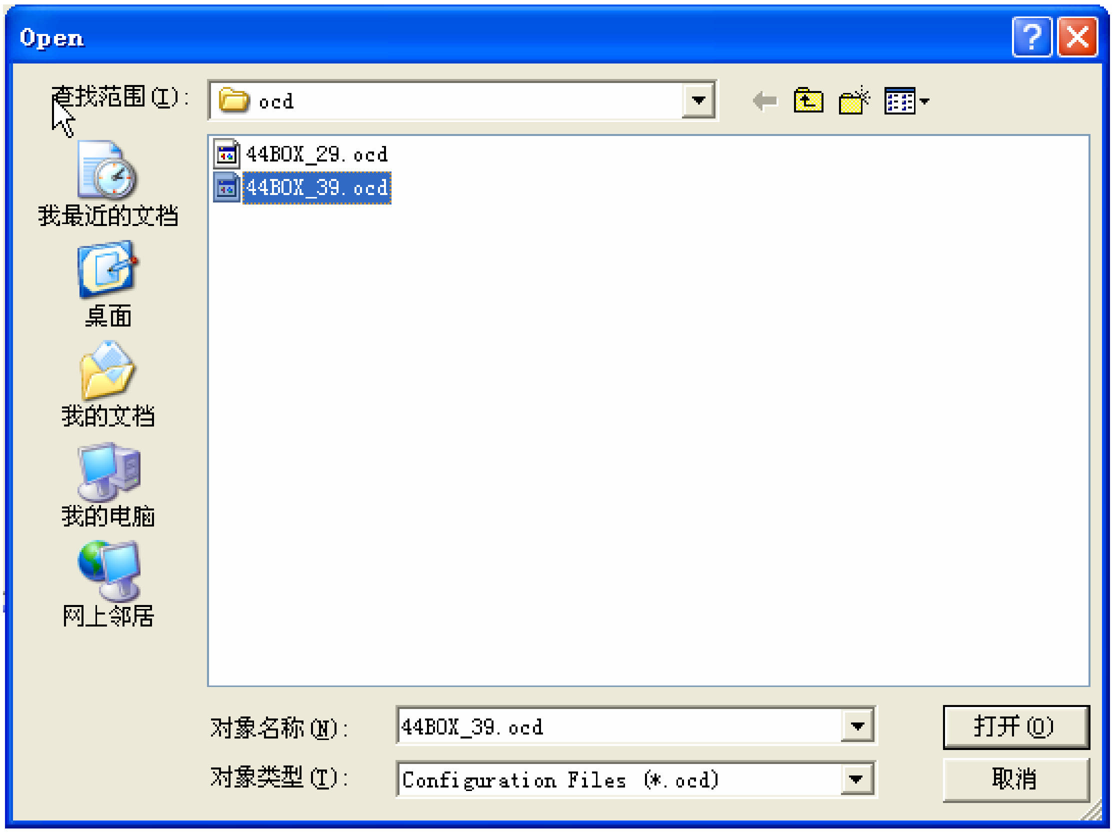
然后编译就会生成你所设置文件名的 ELF 格式文件了。

注意：Flashpgm 烧写时 Jtag 的接法。（Jtag 小板上选择 wiggler 接口）

5.4.1.2 Flashpgm 的设置

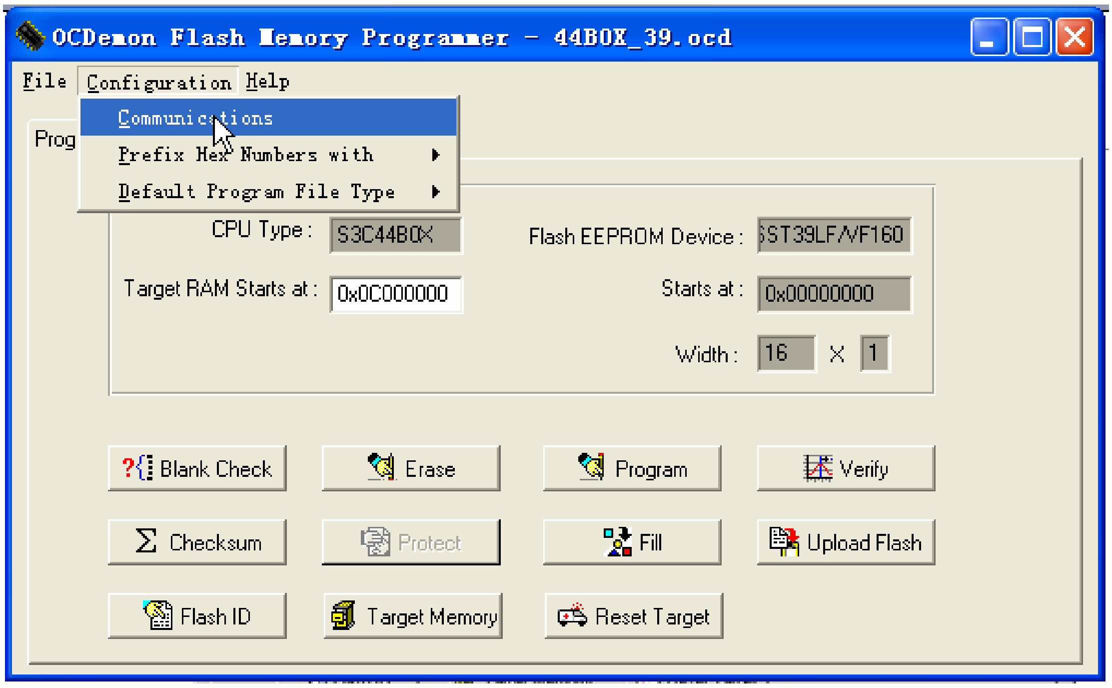
调入 44B0X_39.ocd 文件

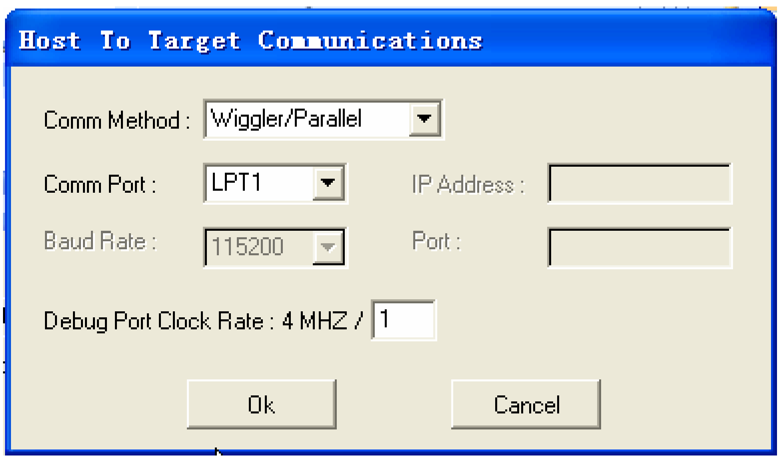




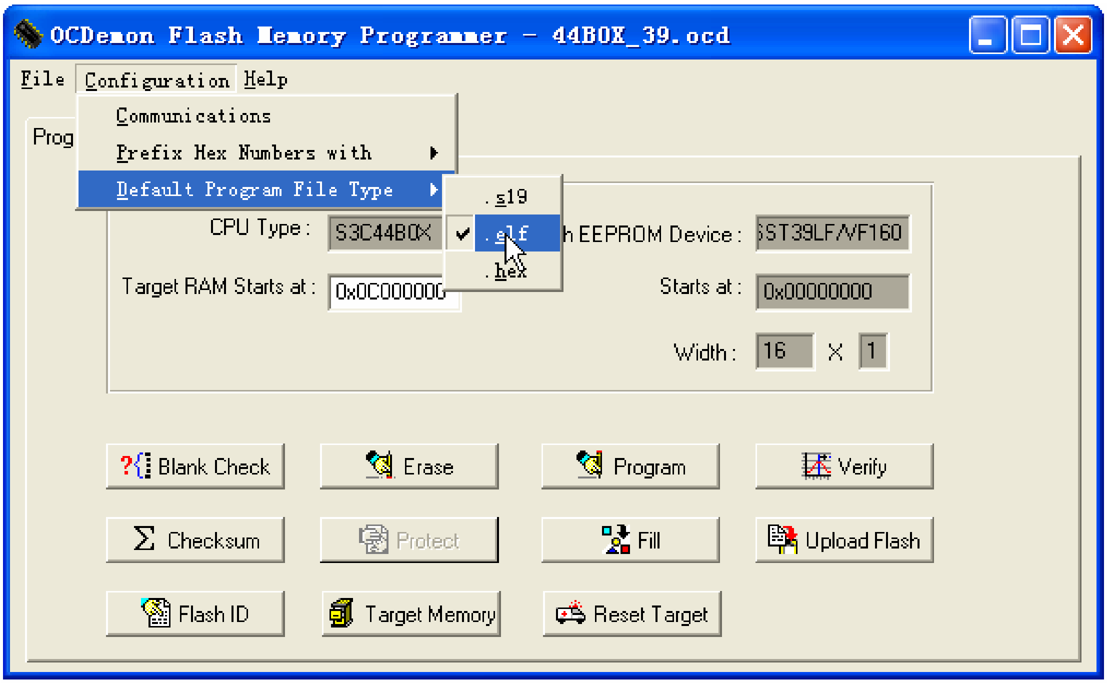
5.4.1.3 设置通信端口

选择 Wiggler Parallel ：

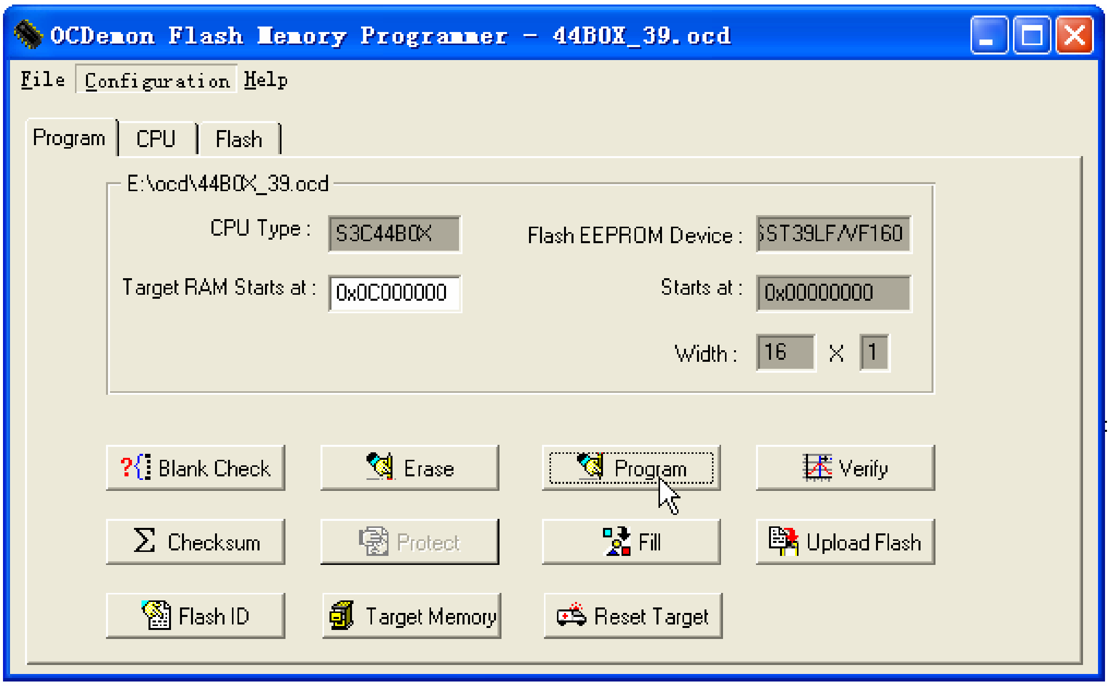




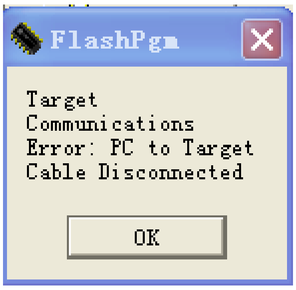
5.4.1.4 选择编程文件类型



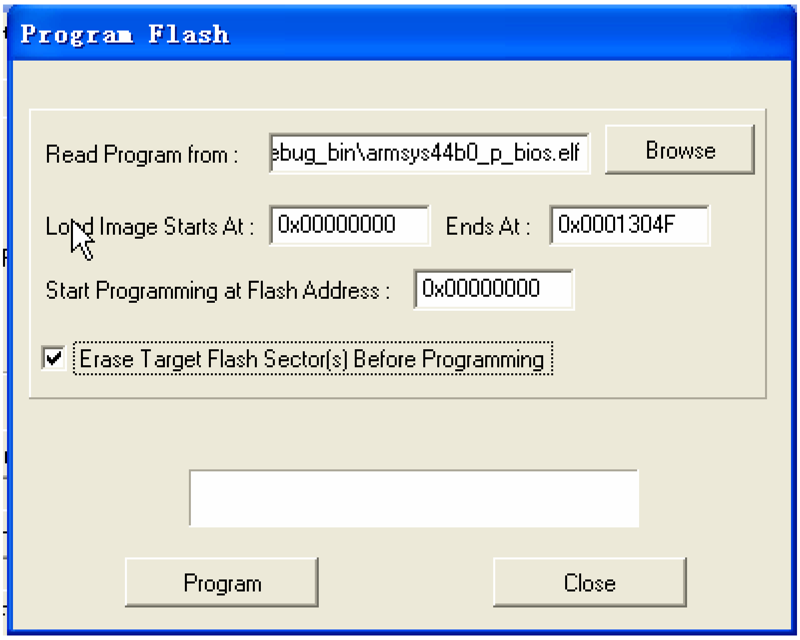
5.4.1.5 点击编程按钮

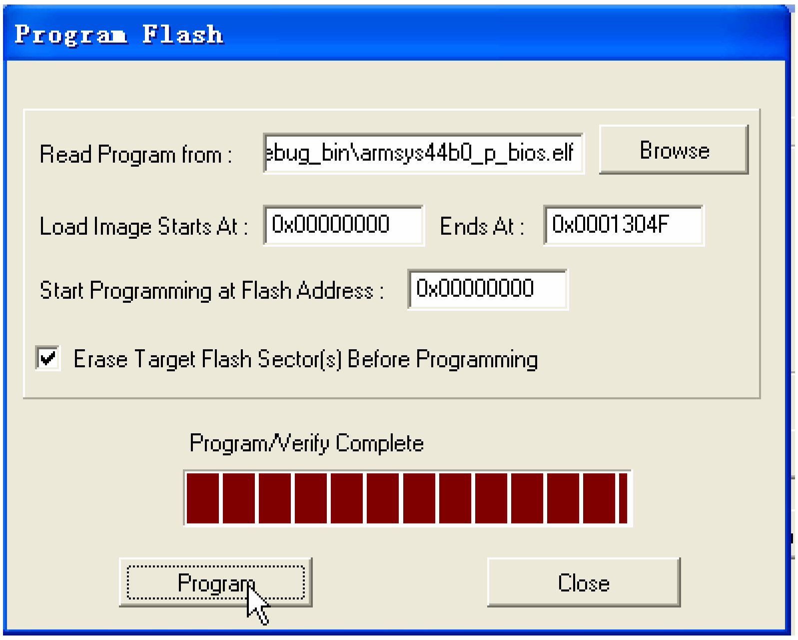


如果目标板连接不正常将显示如下对话框，请检查目标板电源是否打开，20 芯排线是否连接正常。



如果连接正常，将显示如下对话框。点击 Browse 按钮，找到要烧写的文件的位置，如下图，另外，我们也要选中 Erase Target Flash Sector(s) Before Program





如果在上面的画面中，你发现按键 Program 始终为灰色，无论如何也不行。请

点击 Close 按键关闭上面的画面，然后按一下板子上面的复位按键，再点击 Flashpgm 界面中的 Program 按键，重新进入上面的画面，就会发现 Program 不在是灰色的了，这样就可以编程了。这实际上是 Flashgpm 的一个 bug，但没有更好的办法，不过这种情况不多见而已。

5.4.2 用 programmer 工程烧写

programmer 是一个烧写工程，用它进行烧录要使用到 ADS1.2 的 AXD 调试功能。programmer 烧录的优点是速度较 Fluted 快。操作步骤如下：

将开发板配套光盘中“开发工具\programmmer\ads_programmer”文件夹拷贝到计算机硬盘中并去掉只读属性。编译该工程进入 AXD 调试环境，然后加载要烧写的启动代码，在 AXD 调试窗口下点击菜单“File-> Load Memory From File…”或者点击图标出现下图所示窗口：

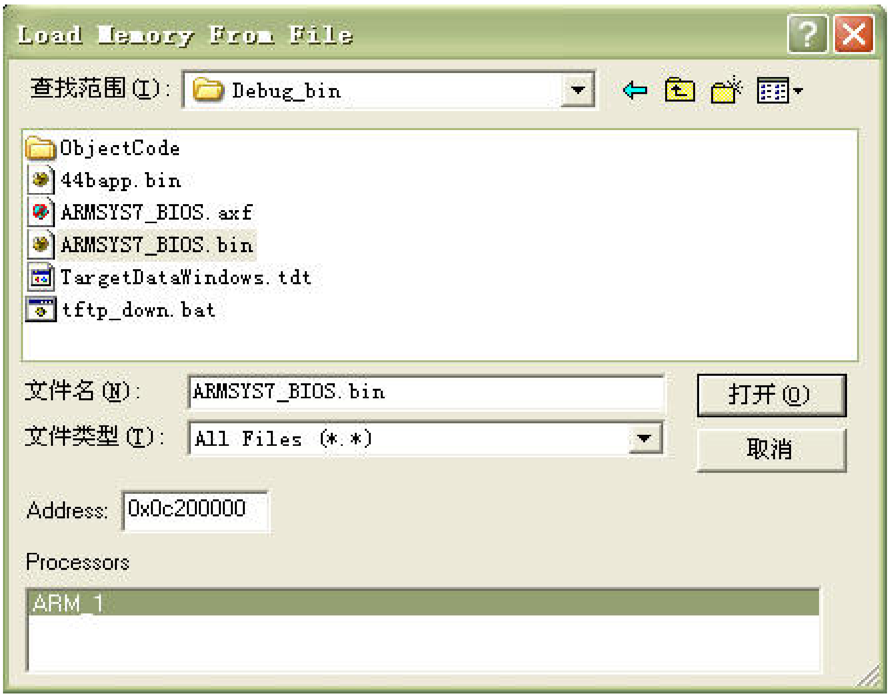


图 6－8 programmer 烧写工程调试时启动代码的加载

选择要烧写的代码文件（如 ARMSYS44B0_P_BIOS.bin），在 Address 地址栏中输入“0x0c200000”，然后点击打开，可以看 JTAG 模块加载代码时的绿色工作发光管被点亮，AXD 调试环境中也可看到加载进度调条。

这样启动代码 bin 文件很快被下载到相应的地址。全速运行烧录程序。可以通过超级终端观察到烧录的过程，烧录过程分为片擦除、空检查、写入和校验，烧录成功的输出信息为：

```
Chip erased!
Blank check OK!
Begin to Write flash...
Write OK! Begin to Verify...
Verify OK!
```

如果终端输出以上信息，其间没有报告错误，说明烧录已经成功了。

第六章 ARMSYS44B0-P 测试程序

6.1 测试前的准备

首先请仔细阅读以上第三章和第五章的说明书，并在全面测试开发板的各项功能之前进行如下准备工作：

（1）将配套的+5VDC 开关电源连接到开发板。

（2）用串口电缆连接开发板的 UART0 接口到计算机串口。

（3）用交叉对等网线（套件中提供的）连接开发板和计算机的 RJ45 接口，或者用普通直连网线将开发板连接到计算机所在的局域网中。

（4）准备一 PS/2 接口的键盘，接至开发板的 PS/2 接口。

（5）准备一张 SD 卡或者是 MMC 卡，插入 SD 卡座中。

（6）准备一副耳机或小音箱，接入 LINE-OUT 音频输出座。

（7）准备一 USB 鼠标和 U 盘，测试 SL811 主从 USB 芯片时要用到。

（8）如有液晶屏和触摸屏模块时，用排线接至 LCD&TSP 接口。（根据液晶屏工作电源要求选择 JP2 的跳线位置）

（9）拨动电源转换开关 SW1 至“EXT”位置，此时开发板接通电源开始工作，红色电源指示发光管点亮，有一绿色发光管规律闪烁。

6.2 ARMSYS44B0-P 功能部件测试

开发板电源打开后，在超级终端窗口中可以看到开发板启动信息。然后输入“test”命令开始执行测试程序，如下图所示：

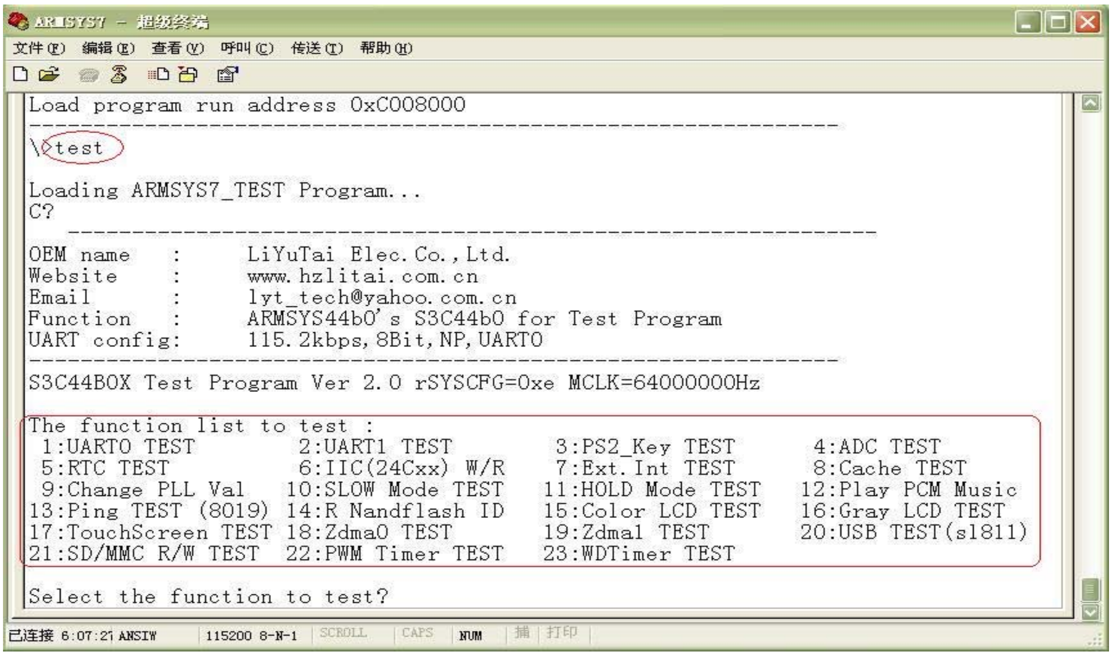
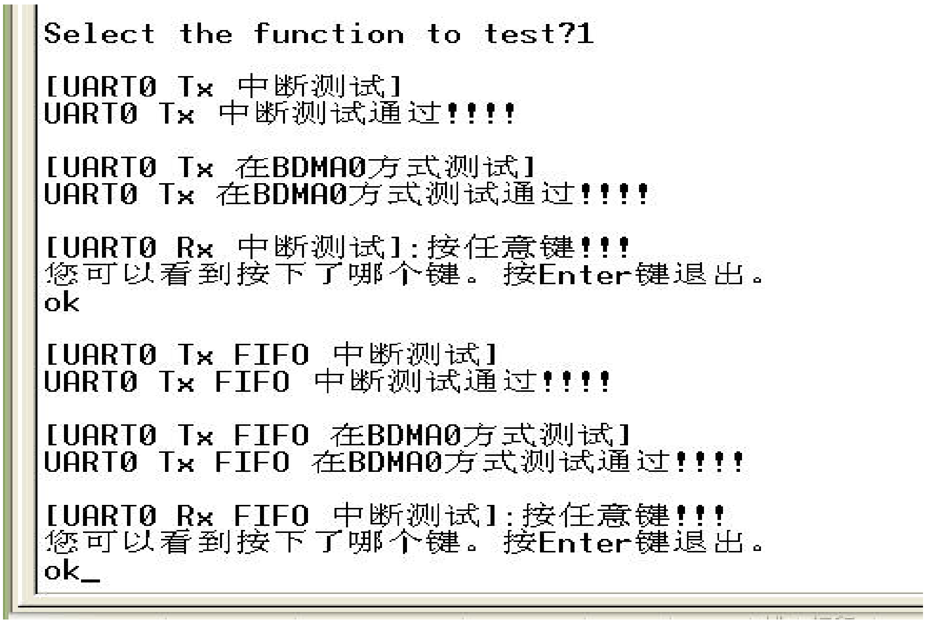


图 6－1 启动测试程序

6.2.1 串口 UART0 测试

在超级终端中输入数字“1”回车，开始执行串口 UART0 的测试，如下图所示：



```
Select the function to test?1

[UART0 Tx 中断测试]
UART0 Tx 中断测试通过!!!!

[UART0 Tx 在BDMA0方式测试]
UART0 Tx 在BDMA0方式测试通过!!!!

[UART0 Rx 中断测试]:按任意键!!!
您可以看到按下了哪个键。按Enter键退出。
ok

[UART0 Tx FIFO 中断测试]
UART0 Tx FIFO 中断测试通过!!!!

[UART0 Tx FIFO 在BDMA0方式测试]
UART0 Tx FIFO 在BDMA0方式测试通过!!!!

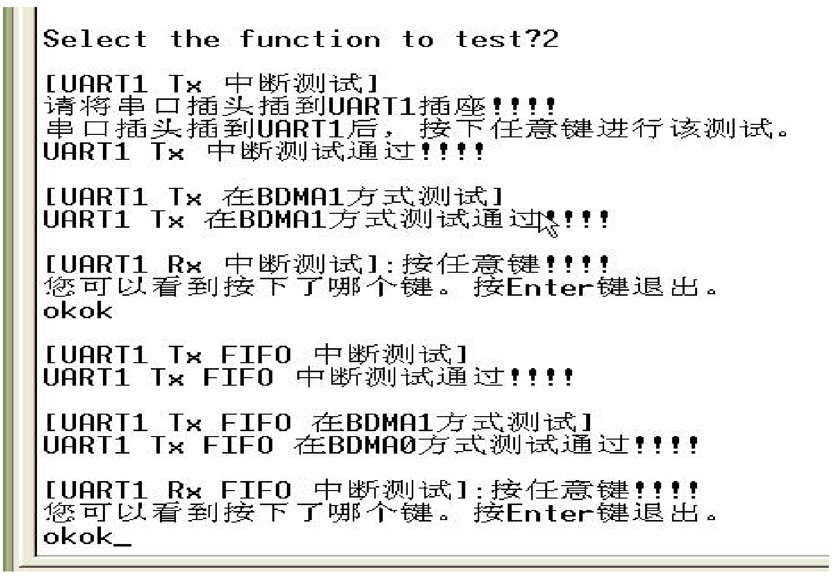
[UART0 Rx FIFO 中断测试]:按任意键!!!
您可以看到按下了哪个键。按Enter键退出。
ok_
```

图 6－2 串口 UART0 测试结果

在串口测试中根据提示信息一步步向下测试。在最后一行“回车”后退出该测试项目。

6.2.2 串口 UART1 测试

当在超级终端中显示选择测试时输入“2”回车，开始串口 UART1 测试，如下图所示：



```
Select the function to test?2

[UART1 Tx 中断测试]
请将串口插头插到UART1插座!!!!
串口插头插到UART1后，按下任意键进行该测试。
UART1 Tx 中断测试通过!!!!

[UART1 Tx 在BDMA1方式测试]
UART1 Tx 在BDMA1方式测试通过!!!!

[UART1 Rx 中断测试]:按任意键!!!!
您可以看到按下了哪个键。按Enter键退出。
okok

[UART1 Tx FIFO 中断测试]
UART1 Tx FIFO 中断测试通过!!!!

[UART1 Tx FIFO 在BDMA1方式测试]
UART1 Tx FIFO 在BDMA0方式测试通过!!!!

[UART1 Rx FIFO 中断测试]:按任意键!!!!
您可以看到按下了哪个键。按Enter键退出。
okok_
```

图 6－3 串口 UART1 测试结果

6.2.3 PS/2 键盘接口测试

当在超级终端中显示选择测试时输入“3”回车，开始 PS/2 接口的测试，该测试是接一 PS/2 接口的键盘。测试时，在外接的键盘上敲击数字键和字母键可以在超级终端上显示出所敲的键，敲外接键盘的回车键退出该测试项，测试过程和结果如下图所示：

```
Select the function to test?3

[PS/2 KeyBoard Test]
Wait KeyBoard key...Enter for back!
123456789F1F2F3F4F5F60.
The function list to test :
 1:UART0 TEST          2:UART1 TEST          3:PS2_Key TEST        4:ADC TEST
 5:RTC TEST            6:IIC(24Cxx) W/R      7:Ext.Int TEST        8:Cache TEST
 9:Change PLL Val     10:SLOW Mode TEST     11:HOLD Mode TEST     12:Play PCM Music
13:Ping TEST (8019)   14:R Nandflash ID     15:Color LCD TEST     16:Gray LCD TEST
17:TouchScreen TEST   18:Zdma0 TEST         19:Zdma1 TEST         20:USB TEST(sl811)
21:SD/MMC R/W TEST

Select the function to test?

已连接 0:11:53 ANSIW   115200 8-N-1   SCROLL   CAPS   NUM   捕   打印
```

图 6—4 外接 PS/2 键盘接口的测试

6.2.4 模数转换 A/D 测试

当在超级终端中显示选择测试时输入“4”回车，开始模数转换 A/D 测试，由于没有输入模拟信号所以测试输出为近似值。如下图所示：

```
Select the function to test?4

现在开始A/D转换自检测试!!!!
按任意键退出!!!!
ADC转换频率=7812(Hz)

通道0:0512  通道1:0479  通道2:0495  通道3:0495
通道4:0495  通道5:0495  通道6:0495  通道7:0511

通道0:0512  通道1:0495  通道2:0495  通道3:0495
通道4:0495  通道5:0495  通道6:0511  通道7:0512

通道0:0479  通道1:0495  通道2:0495  通道3:0495
通道4:0495  通道5:0511  通道6:0512  通道7:0511
```

图 6—5 模数转换 A/D 测试

6.2.5 实时时钟 RTC 测试

当在超级终端中显示选择测试时输入“5”回车，开始实时时钟 RTC 测试。测试时，可以在超级终端上观察到实时时钟在准确地走动，时钟走 10 秒钟后自动退出测试。测试结果如下图所示：

```
Select the function to test?5


RTC测试程序运行中，请稍后...
RTC定时报警中断设定时间为：10秒
RTC节拍中断测试...
2005年 4月20日,星期三, 9:16:12
2005年 4月20日,星期三, 9:16:12
RTC定时报警中断测试通过!!!!
RTC节拍中断测试通过!!!!
```

图 6－6 实时时钟 RTC 测试结果

6.2.6 IIC 存储器（AT24C04）测试

当在超级终端中显示选择测试时输入“6”回车，开始 IIC 存储器 AT24C04 的读写测试。该测试是先通过 IIC 接口写如数据到 24C04，然后再读出数据显示在超级终端上。测试结果如图所示：

```
[IIC Test using AT24Cxx]
Write test data into AT24LCxx
Read test data from AT24LCxx
 0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f
20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f
30 31 32 33 34 35 36 37 38 39 3a 3b 3c 3d 3e 3f
40 41 42 43 44 45 46 47 48 49 4a 4b 4c 4d 4e 4f
50 51 52 53 54 55 56 57 58 59 5a 5b 5c 5d 5e 5f
60 61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f
70 71 72 73 74 75 76 77 78 79 7a 7b 7c 7d 7e 7f
80 81 82 83 84 85 86 87 88 89 8a 8b 8c 8d 8e 8f
90 91 92 93 94 95 96 97 98 99 9a 9b 9c 9d 9e 9f
a0 a1 a2 a3 a4 a5 a6 a7 a8 a9 aa ab ac ad ae af
b0 b1 b2 b3 b4 b5 b6 b7 b8 b9 ba bb bc bd be bf
c0 c1 c2 c3 c4 c5 c6 c7 c8 c9 ca cb cc cd ce cf
d0 d1 d2 d3 d4 d5 d6 d7 d8 d9 da db dc dd de df
e0 e1 e2 e3 e4 e5 e6 e7 e8 e9 ea eb ec ed ee ef
f0 f1 f2 f3 f4 f5 f6 f7 f8 f9 fa fb fc fd fe ff
```

图 6－7 IIC 存储器读写测试

6.2.7 外部中断测试

当在超级终端中显示选择测试时输入“7”回车，开始外部中断测试。测试时，按下开发板上的四个外部中断按键，对应显示出那个中断键被按下了，测试完后按计算机键盘的任意键退出该测试。测试结果如图所示：

```
[External Interrupt Test]
Note:Because Keyboard's pressing exist shake,many times
of interrupt maybe triggered.

Using Fallling trigger.Press the EINT buttons!!!

EINT4 had been occured...

EINT5 had been occured...

EINT6 had been occured...

EINT7 had been occured...
```

图 6－8 外部中断按键 Exint4—Exint7 测试

6.2.8 IIS 音频播放 WAV 文件测试

要开始本项测试之前首先要读取或下载 WAV 文件到“0xC200000”地址，可以从 NandFlash 中读取文件名为“sound”的 WAV 文件，也可从计算机下载自己的 WAV 格式的文件。将 WAV 格式的音频文件读取或下载后，在超级终端中显示选择测试时输入“12”回车，开始 IIS 音频测试。测试时，可以在插到 LINE-OUT 音频座的耳机或音箱中听到播放的 WAV 音频文件内容。在超级终端中观测到的结果如图所示：

```
Select the function to test?12¶
[UDA1341 test for Play PCM Music]
rNCACHBE0=0xc400c200
sample size=0x12e12

[]Now play 0x0c200000 the wave file...
Push any key to exit!!!
.......Exit Playing.
```

图 6－9 IIS 音频播放 WAV 文件测试

6.2.9 以太网卡 8019 测试

在该项测试之前首先要将网线连接良好，设置计算机的 IP 地址为“192.168.253.x”（x 不能为 2, 192.168.253.2 是开发板本身的 IP 地址），还要将计算机的所有防火墙关闭，然后再开始以太网测试。该测试主要是通过 ping 工具来测试网络芯片工作是否正常，正常工作时会有三次正确的发送和接收的过程。

当在超级终端中显示选择测试时输入“13”回车，开始以太网测试，这时要输入如图所示的“ping –v 192.168.253.x”（x 除了 2）命令后回车。

```
Select the function to test?13


PING v0.19
Enter Command line:(Example, ping 192.168.253.3)
ping -v 192.168.253.23_
```

图 6－10 以太网测试输入的 ping 命令

其中-v 表示解析整个 ping 过程，192.168.253.x 表示你的 PC 机的 IP 地址。如果以太网工作正常，超级终端上应出现如下显示:

```
Reply from J seq=256 len=32 OK

Tx0 /len 74 00:0e:a6:08:09:e0 00:0e:a6:08:09:e0IP 00:0e:a6:08:09:e0 00:0e:a6:08:
09:e0-> 00:0e:a6:08:09:e0 ICMP
Rx0 \len 74 00:0e:a6:08:09:e0 00:0e:a6:08:09:e0IP 00:0e:a6:08:09:e0 00:0e:a6:08:
09:e0-> 00:0e:a6:08:09:e0 ICMPJ
Reply from J seq=512 len=32 OK

Tx0 /len 74 00:0e:a6:08:09:e0 00:0e:a6:08:09:e0IP 00:0e:a6:08:09:e0 00:0e:a6:08:
09:e0-> 00:0e:a6:08:09:e0 ICMP
Rx0 \len 74 00:0e:a6:08:09:e0 00:0e:a6:08:09:e0IP 00:0e:a6:08:09:e0 00:0e:a6:08:
09:e0-> 00:0e:a6:08:09:e0 ICMPJ
Reply from J seq=768 len=32 OK
ICMP echo: 3 sent, 3 received, 0 errors

The function list to test :
 1:UART0 TEST         2:UART1 TEST         3:PS2_Key TEST       4:ADC TEST
 5:RTC TEST           6:IIC(24Cxx) W/R     7:Ext.Int TEST       8:Cache TEST
 9:Change PLL Val    10:SLOW Mode TEST    11:HOLD Mode TEST    12:Play PCM Music
13:Ping TEST (8019)  14:R Nandflash ID    15:Color LCD TEST    16:Gray LCD TEST
17:TouchScreen TEST  18:Zdma0 TEST        19:Zdma1 TEST        20:USB TEST(sl811)
21:SD/MMC R/W TEST

Select the function to test?_
```

已连接 1:41:51 ANSIW 115200 8-N-1 SCROLL CAPS NUM 捕 打印

图 6－11 以太网测试结果

6.2.10 NandFlash 读取 ID 号测试

当在超级终端中显示选择测试时输入“14”回车，此时超级终端上会出现如下图所示的信息。

```
Select the function to test?14

NandFLASH Chip's ID is 0xec73
```

图 6－12 显示 NandFlash 芯片 ID

6.2.11 彩色 LCD 的测试

在开始该项测试之前，要先将与开发板配套的彩色液晶连接到开发板上，液晶的工作电源选择正确（JP2 跳线）。在超级终端中显示选择测试时输入“15”回车，开始 STN256 色 LCD 的测试，在超级终端上看到如图所示内容，在彩色 LCD 上可以看到显示各种不同的颜色。等到该测试程序执行完后自己退出。

```
Select the function to test?15

STN LCD TEST: (SHARP 5.7' 256C 320*240P)
_
```

图 6－13 彩色 LCD 测试

6.2.12 16 级灰度 LCD 的测试

如果与开发板配套的是 16 级灰度液晶就选择该项测试，在测试之前也要将液晶连接到开发板上并选择好液晶电源跳线（JP2），然后在超级终端中显示选择测试时输入“16”回车，在超级终端上显示如图所示信息，在液晶屏上显示测试的过程。

```
Select the function to test?16

Monochrome LCD TEST: (CASIO 3.8' 16G 320*240P)
_
```

图 6－14　16 级灰度液晶测试

6.2.13 触摸屏（Touch Screen Panel）测试

将开发板配套的触摸屏模块连接到开发板上，在超级终端中显示选择测试时输入“17”回车，开始触摸屏的测试，此时超级终端上显示如图所示：

```
Select the function to test?17

[TOUCH SCREEN PANEL TEST]
Press any point on the TOUCH SCREEN PANEL...
Push any key on PC keyboard to Exit!
_
```

图 6－15 触摸屏测试

轻轻点击触摸屏的不同位置，可以在超级终端上显示如图所示信息：

```
Touch at the point (  234,  226 )
Touch at the point (   21,  229 )
Touch at the point (   21,   31 )
Touch at the point (  233,   33 )
Touch at the point (  119,  136 )
```

图 6－16 点击触摸屏的位置坐标

6.2.14 USB－HOST（SL811HST）测试

进行该测试之前要将开发板上的跳线（JP1）设为“master”，在超级终端中显示选择测试时输入“20”回车，开始用 SL811HST 组成的主 USB 的测试。此时在超级终端上显示如图所示内容，根据提示内容插入 USB 从设备点任意键开始对 USB 设备的配置，在超级终端上可以看到配置的过程。

```
Select the function to test?20

SL811HST Host MODE test.
Please insert a USB Device!
For example : USB Flash Disk, USB MOUSE, USB keyboard...

Please insert a USB Device!
Press ESC key to Exit, other key to continue...
```

图 6－16 主 USB 测试

出现以上信息后插入 USB 从设备，然后按任意键开始对 USB 设备的配置，配置过程入图所示：

```
DNW v0.50A [COM1,115200bps][USB:x]
Serial Port USB Port Configuration Help
EP0 Status 40
EP0 Status 9
BR(4,3,9,4,)
EP0 Status 1
BW[80,6,1,3,9,4,4,0,]
EP0 Status 1
EP0 Status 40
EP0 Status 40
EP0 Status 9
BR(e,3,41,0,)
EP0 Status 1
BW[80,6,1,3,9,4,e,0,]
EP0 Status 1
EP0 Status 40
EP0 Status 40
EP0 Status 9
BR(e,3,41,0,4f,0,4d,0,45,0,47,0,41,0,)
EP0 Status 1
BW[80,6,0,2,0,0,8,0,]
EP0 Status 1
EP0 Status 40
EP0 Status 40
EP0 Status 9
BR(9,2,20,0,1,1,0,80,)
EP0 Status 1
BW[80,6,0,2,0,0,20,0,]
EP0 Status 1
EP0 Status 40
EP0 Status 40
EP0 Status 9
BR(9,2,20,0,1,1,0,80,32,9,4,0,0,2,8,5,)
EP0 Status 1
BR(50,0,7,5,1,2,40,0,1,7,5,82,2,40,0,1,)
EP0 Status 1
Set_Configuration end
EndPoint 0x1, attr = 0x2, pkt_size = 0x40, interval = 0x1
EndPoint 0x82, attr = 0x2, pkt_size = 0x40, interval = 0x1
EnumUsbDev Return
'Full' Speed Device Attached.
```

图 6－17 对 USB 设备的配置过程

6.2.15 SD/MMC 卡读写测试

准备一张 SD 卡或 MMC 卡插入到开发板的 SD 卡座，在超级终端中显示选择测试时输入“21”回车，开始对所插入卡进行读写测试。根据超级终端上的提示输入要读写的扇区数，然后回车后出现下图所示信息：

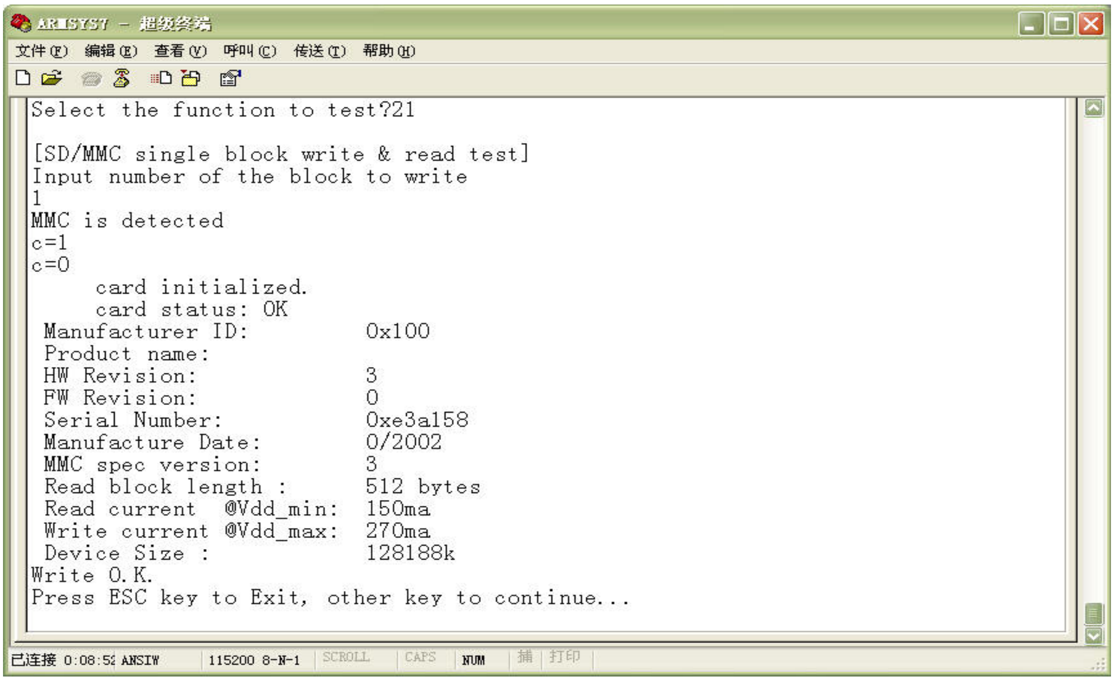


图 6－18 MMC 卡相关信息

出现上图所示时，按任意键继续，按“ESC”键退出测试。下一步读取并显示出刚才写入扇区的数据，如图所示：

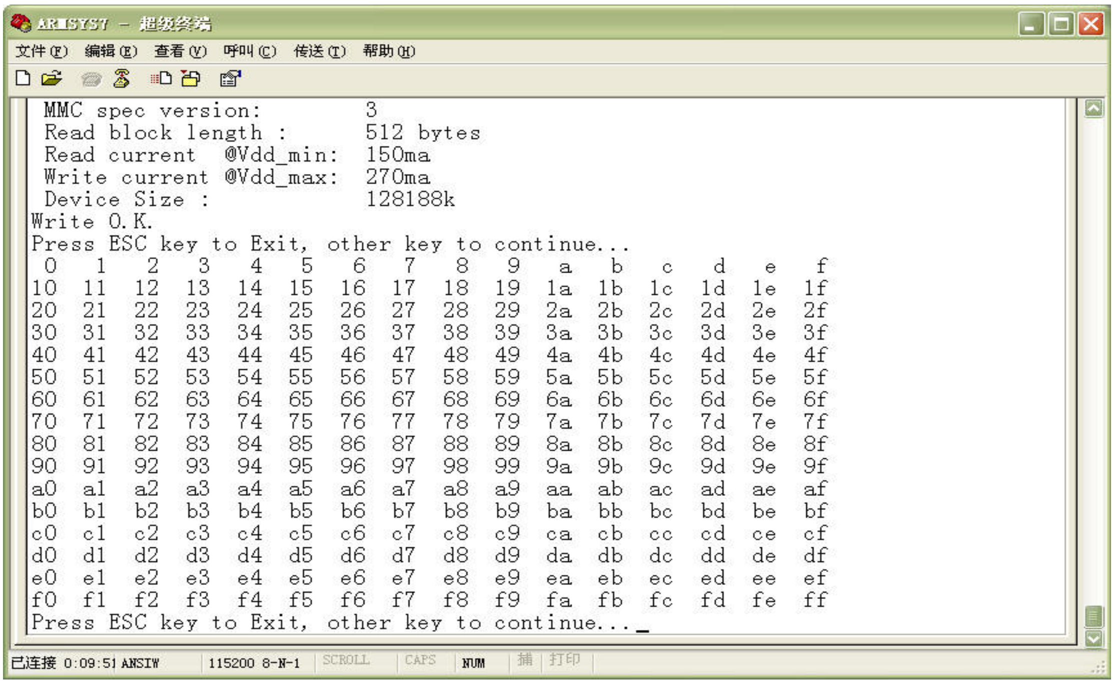


图 6－19 MMC 卡读取指定扇区的数据（前 256 字节）

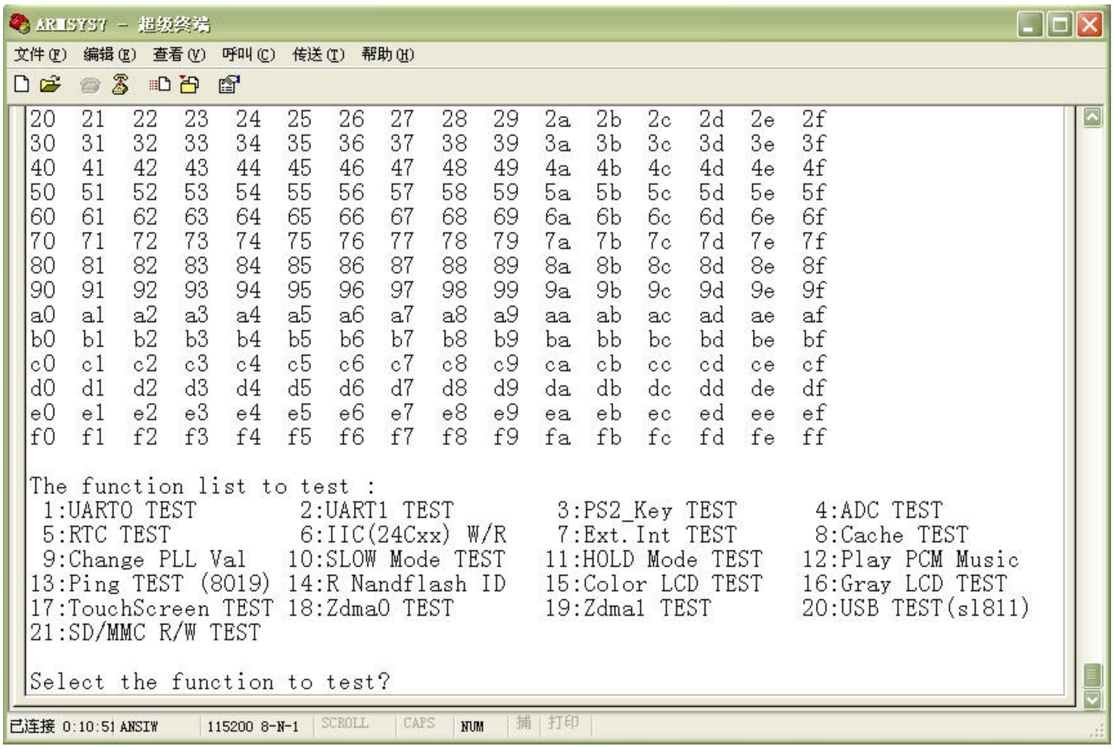


图 6－19 MMC 卡读取指定扇区的数据（后 256 字节）

6.2.16 PWM 定时器测试

当在超级终端中显示选择测试时输入“22”回车，开始 PWM Timer 定时器测试。该测试分为两部分，一部分是针对 5 个定时器定时测试，另一部分是针对 PWM 定时器输出测试。测试结果如图所示：

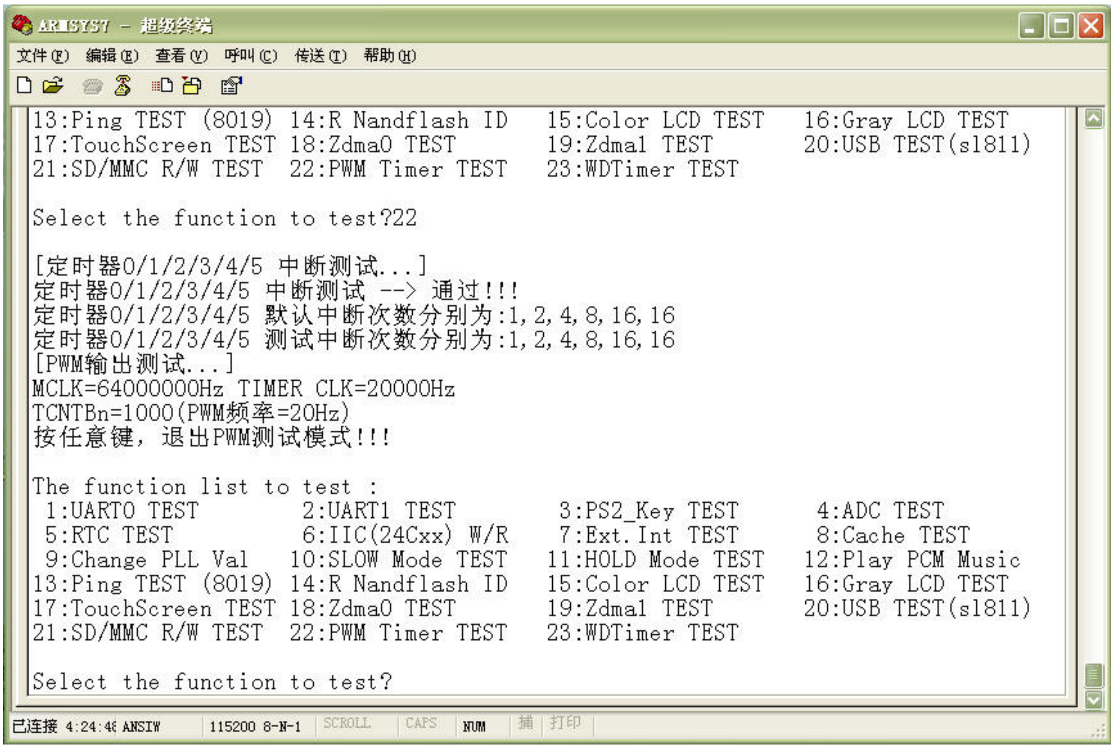


图 6－20 PWM Timer 定时器测试结果

6.2.17 WatchDog Timer 看门狗定时器测试

当在超级终端中显示选择测试时输入“23”回车，开始 WDTimer 看门狗定时器测试。测试结果如图所示：

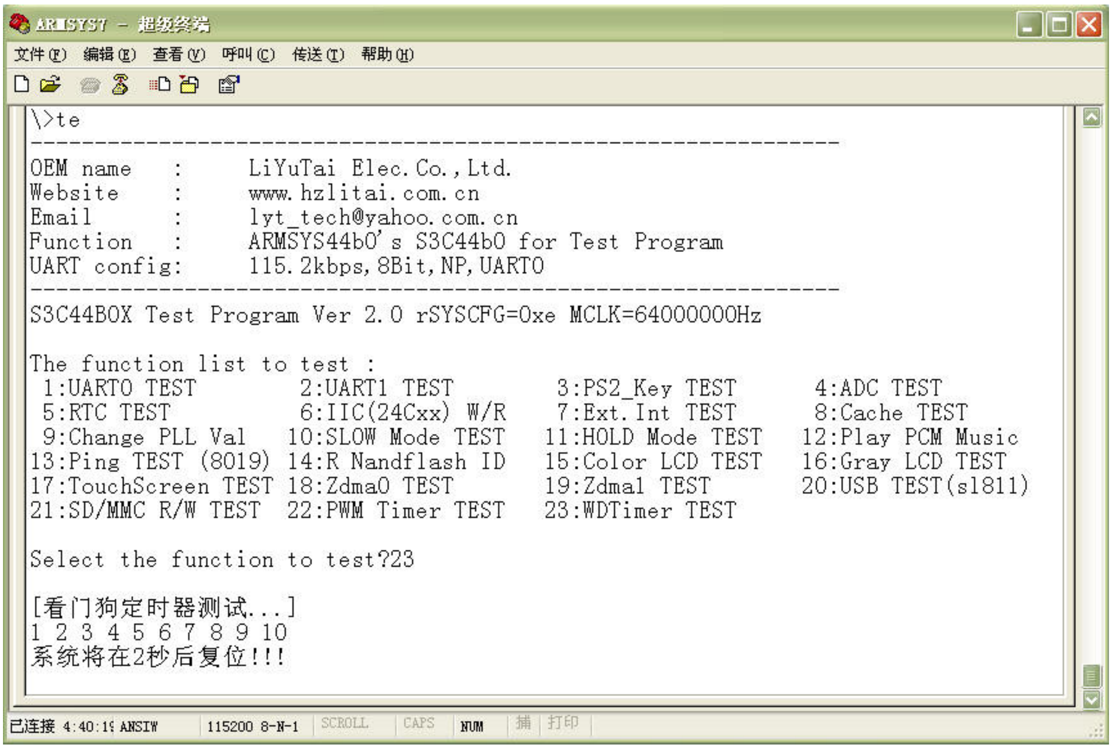


图 6－21 看门狗定时器 WDTimer 测试结果

第七章 uCLinux 的使用说明

7.1 uCLinux 简介

uCLinux 是专门用于微控制领域的嵌入式 Linux 操作系统，它已被成功移植到多种平台上运行，它是针对没有 MMU 的处理器而设计的，uCLinux 符合 GNU/GPL 公约完全开放源代码，它采用 romfs 文件系统体积小巧精简，深受广大微控制领域的用户喜爱， uCLinux 的官方网站:http://www.uClinux.org。

7.2 下载和运行 uCLinux

7.2.1 uCLinux 运行环境的建立

运行 uCLinux 之前要有程序来加载其内核，bootloader 就是完成在操作系统内核运行之前运行的一段小程序。通过这段小程序，我们可以初始化硬件设备、建立内存空间的映射图，从而将系统的软硬件环境带到一个合适的状态，以便为最终调用操作系统内核准备好正确的环境。ARMSYS44B0-P 开发板用的 BIOS 中已经包含了 bootloader 代码，在开发板 BIOS 运行后可直接将 uCLinux 内核下载运行，或者直接在开发板启动后输入命令“uclinux”即可加载 uCLinux 到内存并运行。

编译好的 uCLinux 内核可以存放在 NorFlash 中，也可以用“uclinux”为文件名存放到 NandFlash 中，开发板的 BIOS 中的“uclinux”命令会自动寻找的。因此 ARMSYS44B0-P 开发板给 uCLinux 提供了较完善的 bootloader，用户个可以灵活运用。

7.2.2 uCLinux 内核的存储

1．存储到 Nand Flash 中

这是比较方便的存储方法，可以随时更改保存。保存过程就是先将编译好的 uCLinux 内核下载到 SDRAM 中，然后用 BIOS 中的“nfw uclinux”命令将其写入即可，那么内核就会长期保存在开发板上。

2．存储到 Nor Flash 中

如果要存储 uCLinux 内核到 NorFlash 中，ARMSYS44B0-P 开发板提供了烧写工具，使写入存储变简化些。烧写工具是前面提到过的 44bapp.bin 程序代码，它可以完成同时将开发板的 BIOS 代码、测试程序代码和 uCLinux 内核烧写到 NorFlash 中，具体步骤如下：

（1）将烧写工具 44bapp.bin 通过网口或串口下载到地址 0xC008000 中。

（2） 将开发板配套的 ARMSYS44B0_P_BIOS.bin 代码文件下载到地址 0xC200000 起始位置。（作为 bootloader 使用）

（3） 将开发板配套的 ARMSYS44B0_P_TEST.bin 代码文件下载到地址 0xC210000 起始位置中。

（4）将编译好的 uCLinux 内核文件代码 imagerom.bin 下载到起始地址为 0xC230000 处。

（5）以上步骤顺利完成后就可以开始烧写了，在超级终端中输入运行命令“run c008000”，这时程序会提示用户是否“运行指定地址中的程序”输入“y”，下面还会有提示判断“确认烧录吗？（y/n）”，输入“y”烧写工具开始运行，四个绿色用户指示灯会循环点亮，说明烧写程序还在进行，直到四个绿色发光管全部点亮或超级终端上显示烧写完成，这时说明烧写工具已经将 BIOS、测试程序和 uCLinux 内核烧写到了 NorFlash 中。然后按下复位按键复位开发板，只要 BIOS 顺利启动开发板，说明以上烧写基本完成，在超级终端中输入命令“test”和“uclinux”，分别运行测试程序和启动 uCLinux，如果两者运行正常说明烧写成功。如果出现异常说明烧写不完全，查找原因重新烧写。

7.2.3 运行 uCLinux

在开发板配套光盘中的“uCLinux/images”目录下，有 2 个 bin 类型文件，这 2 个文件就是 uCLinux 源码包经过交叉编译后产生的二进制文件，可以用来下载或烧录到开发平台上运行。这 2 个文件的原文件名为 image.ram，image.rom ，由于我们要进行下载，因此改文件名为 imageram.bin 和 imagerom.bin。这两个文件的作用是：其中 imageram.bin 文件，是未经压缩的 uCLinux 内核加 root 文件系统的二进制文件，可以直接下载到 SDRAM 的指定地址（0xc008000）处运行；imagerom.bin,是经过压缩的内核加 root 文件系统二进制文件，可以下载到 SDRAM 中的指定地址（0xc100000）处，自行解压缩运行；还可以被烧录到 flash ROM 的 0x200000 地址处，启动 uClinux 时，由 Bootloader 拷贝到 SDRAM 再解压运行。

7.3 配置和编译 uCLinux

下面的工作——对 uCLinux 的配置和编译，要在安装了 Linux 操作系统的 PC 机上进行。推荐采用 Redhat9.0。

7.3.1 解压 uClinux 移植包

在开发板配套光盘中提供的 uClinux-ARMSYS-20051111.tar.gz 是移植自 uClinux-dist-20040408.tar.gz，20040408 版本在很多方面比早先的 20030522 版本要完善很多。这里我们使用的内核版本是 Linux 2.4.24。

uClinux-ARMSYS-20051111.tar.gz 是针对 ARMSYS44B0-P 的硬件平台进行移植的，它对 uClinux-dist-20040408.tar.gz 所做的修改部分保存在 uClinux-20040801-ARMSYS.patch 补丁文件中，供用户参考（用户不需要自己打补丁）。

将 uClinux-ARMSYS-20050101.tar.gz 拷贝到/home/ 下，运行解压命令：

“tar xvzf uClinux-ARMSYS-20051111.tar.gz”

解压结束后会在/home/ 下生成 uClinux-dist 目录。

7.3.2 安装编译环境

将 arm-elf-tools-20030314.sh 拷贝到根目录，运行安装：

sh arm-elf-tools-20030314.sh

7.3.3 配置和裁减 uClinux

下面开始配置 uClinux 的内核和用户选项。打开终端。

cd /home/uClinux-dist （进入/home/uClinux-dist 目录下）

make menuconfig （运行 uClinux 配置环境）

进入 uClinux 配置(uClinux v3.1.0 Configuration) ，选中“Kernel/ Library/Defaults Selection ->”敲空格进入。其中有两个选项需要选取：“内核设置和用户选项设置”：

[*] Customize Kernel Settings

[*] Customize Vendor/User Settings

选中这两项，按下 ESC 键退出，在询问是否保存时，选择 Yes 并回车。终端将首先进入内核配置选单。我们在配置 uClinux 内核时，就可以通过对这些选项的选择和取消选择来设定内核所具有的功能项。这也是裁减 uClinux 内核的基本方法。

按下 ESC 退出后将进入用户选项选单。用户选单中所列举的是一些常用工具和用户程序,我们自己编写的应用程序也可以放入用户选单。例如在选单中的：

My new application --->

中就有一个最简单的 Helloworld 应用程序。

用户选项中选中的内容，将交叉编译成压缩格式的可执行二进制文件，放入到 rom 文件系统中供用户调用运行。

了解这些选项之后，可以暂时不对这些选项进行任何修改。按 ESC 键退出配置界面。

7.3.4 编译 uClinux

在/opt/目录下，按下面的步骤对 uClinux 源码包进行编译：

make dep……………………………建立依赖关系

make clean（非必要）……………清除旧的编译结果

make lib_only ……………………编译库

make user_only……………………编译用户程序

make romfs…………………………产生 rom 文件系统

make image…………………………产生映像文件

make………………………………编译内核

编译成功后，在 uClinux-dist/ 目录下将产生 images 目录，其中包含的 2 个文件：image.ram, image.rom 就是我们可以用来下载和烧录的映像文件。然后下载或烧录这些二进制文件，并启动运行 uClinux。

7.4 uCLinux 的应用开发

开发应用程序的简单例子：Helloworld 和跑马灯实例，请查看《参考资料》中的 helloworld_armsys.pdf 文档。

7.4.1 开发模式

ARMSYS 上 uClinux 应用的开发目前支持的调试模式，一种是编译产生了二进制映像文件后，通过 BIOS 所带有的 TFTP 下载到目标板上运行观察运行结果，在开发应用程序部分时，可以采用 NFS，mount 上宿主机的硬盘的某个目录，运行其中的可执行程序。因为 mount 的宿主机硬盘上的应用程序会自动覆盖更新，再重新执行的就是更改后的新版本。这样反复调试、更改、编译再调试，而不必切换操作系统或下载、烧写板子，直至程序工作正常，是一种十分方便的方法。关于 NFS 的移植和使用的详细内容请参考文档：

基于 ARMSYS 的 uClinux 应用之——NFS 开发环境的建立。

注意，uClinux-ARMSYS-20051111.tar.gz 中已经移植好 NFS 了，只需要在配置时添加即可。用户自己开发应用程序一定要放在 uClinux/user 目录下，否则，若放在其它目录下将有许多宏要自己定义，很是繁琐。

7.4.2 调试方法

直接在目标板上调试应用程序有两种办法：

（1）打印串口：

这是嵌入式系统中最常用的调试手段，虽然简单但却有效实用。其实几种方法相比之下，最有效便捷的方法还是 printf。

（2）gdb 调试

目前暂不支持。用户也可以自己在板子上移植 gdbserver。

7.5 在 Linux 操作系统下程序代码的下载

7.5.1 使用 minicom 终端仿真程序

在 Linux 操作系统下，minicom 是一个友好易用的串口通信程序。通过它可以监视并控制串行口的信息。在终端下输入命令：“# minicom ”启动该程序。minicom 的操作使用是基于窗口的。要弹出所需功能的窗口，可按 Ctrl+A 键，然后再按各功能键（a-z 或 A-Z）。先按 Ctrl+A 键，再按 z 键，将出现一个帮助

窗口，它提供了所有命令的简述。按 Ctrl+A 键、O 键，就进入了 setup 菜单。改变其中的波特率为 115200bps，不要硬件或软件流控（8N1）。设置调制解调器的 init 和 reset 的项（用 minicom –s 命令，使用其提供的菜单就可以设置）为空项。确保 ARMSYS44B0-P 开发板已经通过串行口同 PC 相连，上电开发板，就可在终端窗口中观测到开发板从串行口发出的启动信息。如果没有显示请检查 minicom 有关设置是否正确。以上步骤正确完成后，一个终端仿真程序就可正常运行了，可以通过串行口对开发板进行各种操作了。

下面就可在串行口终端通过 XMODEM 方式下载程序代码至开发板了。具体操作是这样的：

1、在开发板上执行 XMODEM 下载命令“xmload c008000”(地址可以改变)。

2、在 minicom 终端窗口按 Ctrl+A 键、S 键，就进入发送文件状态。

3、选择传输方式为 XMODEM，然后选择要发送的文件，确认后开始发送文件到串行口，直到发送完毕。

7.5.2 使用 tftpcmd 网络传输

ARMSYS44B0-P 开发板在 Linux 操作系统下也同样支持 TFTP 网络下载，在终端运行 tftpcmd 程序即可。在使用时，首先要将要下载的文件放置在/tftpboot 目录下，然后在终端输入 “tftpcmd” 命令启动 TFTP 传输。利用 TFTP 下载程序代码的具体步骤如下：

1、在开发板上执行 TFTP 下载命令“netload c008000”(c008000 地址用户可根据实际情况改变)，这时开发板进入网络传输等待状态。

2、在终端运行“ping 192.168.253.2”命令，测试网络是否完好。其中的 IP 地址为开发板当前的 IP 地址。测试结束按 Ctrl+C。

3、在终端运行“tftpcmd”命令，进入 TFTP 传输操作命令行。

4、然后输入命令“connect 192.168.253.2”，IP 地址为开发板地址。

5、输入“put image.ram”命令开始传送 image.ram 文件到开发板指定的地址。

6、传输完成后输入“quit”命令退出到终端命令行。

第八章 相关术语解释

8.1 XMODEM 协议

XMODEM 协议是一种使用拨号调制解调器的个人计算机通信中广泛使用的异步文件运输协议。这种协议以 128 字节块的形式传输数据，并且每个块都使用一个校验和过程来进行错误检测。使用 PC 机传送文件的早期协议。在数据传送过程中使用 1 字节的控制序列，以便通讯双方对传送过程进行监视。如果接收方关于一个块的校验和与它在发送方的校验和相同时，接收方就向发送方发送一个认可字节。然而，这种对每个块都进行认可的策略将导致低性能，特别是具有很长传播延迟的卫星连接的情况时，问题更加严重。

使用循环冗余校验的与 XMODEM 相应的一种协议称为 XMODEM--CRC 。还有一种是 XMODEM--1K ，它以 1024 字节一块来传输数据。ZMODEM 是最有效的一个 XMODEM 版本，它不需要对每个块都进行认可。事实上，它只是简单地要求对损坏的块进行重发。ZMODEM 对按块收费的分组交换网络是非常有用的。不需要认可回送分组在很大程度上减少了通信量。

YMODEM 也是一种 XMODEM 的实现。它包括 XMODEM--1K 的所有特征，另外在一次单一会话期间为发送一组文件，增加了批处理文件传输模式。

8.2 Nand Flash 和 Nor Flash 详解

NOR 和 NAND 是现在市场上两种主要的非易失闪存技术。Intd 于 1988 年首先开发出 NOR flash 技术，彻底改变了原先由 EPROM 和 EEPROM 一统天下的局面。紧接着，1989 年，东芝公司发表了 NAND flash 结构，强调降低每比特的成本，更高的性能，并且象磁盘一样可以通过接口轻松升级。但是经过了十多年之后，仍然有相当多的硬件工程师分不清 NOR 和 NAND 闪存。

“flash 存储器”经常可以与“NOR 存储器”互换使用。许多业内人士也搞不清楚 NAND 闪存技术相对于 NOR 技术的优越之处，因为大多数情况下闪存只是用来存储少量的代码，这时 NOR 闪存更适合一些。而 NAND 则是高数据存储密度的理想解决方案。NOR 的特点是芯片内执行（xIP , cxecutc hi Place ），这样应用程序可以直接在 flash 闪存内运行，不必再把代码读到系统 RAM 中。NOR 的传输效率很高，在 1--4MB 的小容量时具有很高的成本效益，但是很低的写入和擦除速度大大影响了它的性能。

NAND 结构能提供极高的单元密度，可以达到高存储密度，并且写入和擦除的速度也很快。应用 NAND 的困难在于 flash 的管理和需要特殊的系统接口。

性能比较：

flash 闪存是非易失存储器，可以对称为块的存储器单元块进行擦写和再编程。任何 flash 器件的写入操作只能在空或已擦除的单元内进行，所以大多数情况下，在进行写入操作之前必须先执行擦除。NAND 器件执行擦除操作是十分简单的，而 NOR 则要求在进行擦除前先要将目标块内所有的位都写为 0。

由于擦除 NOR 器件时是以 64--128KB 的块进行的，执行一个写入／擦除操作的时间为 55ms ，与此相反，擦除 NAND 器件是以 8--32KB 的块进行的，执行相同的操作最多只需要 4ms 。执行擦除时块尺寸的不同进一步拉大了 NOR 和 NADN 之间的性能差距，统计表明，对于给定的一套写入操作（尤其是更新小文件时），更多的擦除操作必须在基于 NOR 的单元中进行。这样，当选择存储解决方案时，设计师必须权衡以下的各项因素。

NOR 的读速度比 NAND 稍快一些。

NAND 的写入速度比 NOR 快很多。

NAND 的 4ms 擦除速度远比 NOR 的 55ms 快。

大多数写入操作需要先进行擦除操作。

NAND 的擦除单元更小，相应的擦除电路更少。

接口差别：

NOR flash 带有 SRAM 接口，有足够的地址引脚来寻址，可以很容易地存取其内部的每一个字节。

NAND 器件使用复杂的 FO 口来串行地存取数据，各个产品或厂商的方法可能各不相同。8 个引脚用来传送控制、地址和数据信息。

NAND 读和写操作采用 512 字节的块，这一点有点像硬盘管理此类操作，很自然地，基于 NAND 的存储器就可以取代硬盘或其他块设备。

容量和成本：

NAND flash 的单元尺寸几乎是 NOR 器件的一半，由于生产过程更为简单，NAND 结构可以在给定的模具尺寸内提供更高的容量，也就相应地降低了价格。

NOR flash 占据了容量为 1--16MB 闪存市场的大部分，而 NAND flash 只是用在 8--128MB 的产品当中，这也说明 NOR 主要应用在代码存储介质中，NAND 适合于数据存储，NAND 在 secure Digital 、Pc cards 和 MMC 存储卡市场上所占份额最大。

可靠性和耐用性：

采用 flahs 介质时一个需要重点考虑的问题是可靠性。对于需要扩展 MTBF 的系统来说，Flash 是非常合适的存储方案。可以从寿命（耐用性）、位交换和坏块处理三个方面来比较 NOR 和 NAND 的可靠性。

寿命（耐用性）：

在 NAND 闪存中每个块的最大擦写次数是一百万次，而 NOR 的擦写次数是十万次。NAND 存储器除了具有 10 比 1 的块擦除周期优势，典型的 NAND 块尺寸要比 NOR 器件小 8 倍，每个 NAND 存储器块在给定的时间内的删除次数要少一些。

位交换：

所有 flash 器件都受位交换现象的困扰。在某些情况下（很少见，NAND 发生的次数要比 NOR 多），一个比特位会发生反转或被报告反转了。一位的变化可能不很明显，但是如果发生在一个关键文件上，这个小小的故障可能导致系统停机。如果只是报告有问题，多读几次就可能解决了。

当然，如果这个位真的改变了，就必须采用错误探测／错误更正(EDC／ECC ）算法。位反转的问题更多见于 NAND 闪存，NAND 的供应商建议使用 NAND 闪存的时候，同时使用 EDC/ECC 算法。

这个问题对于用 NAND 存储多媒体信息时倒不是致命的。当然，如果用本地存储设备来存储操作系统、配置文件或其他敏感信息时，必须使用 EDC/ECC 系

统以确保可靠性。坏块处理 NAND 器件中的坏块是随机分布的。以前也曾有过消除坏块的努力，但发现成品率太低，代价太高，根本不划算。

NAND 器件需要对介质进行初始化扫描以发现坏块，并将坏块标记为不可用。在已制成的器件中，如果通过可靠的方法不能进行这项处理，将导致高故障率。

易于使用：

可以非常直接地使用基于 NOR 的闪存，可以像其他存储器那样连接，并可以在上面直接运行代码。

由于需要 FO 接口，NAND 要复杂得多。各种 NAND 器件的存取方法因厂家而异。在使用 NAND 器件时，必须先写入驱动程序，才能继续执行其他操作。向 NAND 器件写入信息需要相当的技巧，因为设计师绝不能向坏块写入，这就意味着在 NAND 器件上自始至终都必须进行虚拟映射。

软件支持：

当讨论软件支持的时候，应该区别基本的读 / 写 / 擦操作和高一级的用于磁盘仿真和闪存管理算法的软件，包括性能优化。

在 NOR 器件上运行代码不需要任何的软件支持，在 NAND 器件上进行同样操作时，通常需要驱动程序，也就是内存技术驱动程序（MTD），NAND 和 NOR 器件在进行写入和擦除操作时都需要 MTD 。

使用 NOR 器件时所需要的 MTD 要相对少一些，许多厂商都提供用于 NOR 器件的更高级软件，这其中包括 M--System 的 TrueFFS 驱动，该驱动被 Wind River system 、Microsoft 、QNX Soflwal ℃ System 、Sylllbian 和 Intel 等厂商所采用。

驱动还用于对 DiskonChip 产品进行仿真和 NAND 闪存的管理，包括纠错、坏块处理和损耗平衡。

8.3 SDRAM 存储器

与 Flash 存储器相比较，SDRAM 不具有掉电保持数据的特性，但其存取速度大大高于 Flash 存储器，且具有读/写的属性，因此，SDRAM 在系统中主要用作程序的运行空间，数据及堆栈区。当系统启动时，CPU 首先从复位地址 0x0 处读取启动代码，在完成系统的初始化后，程序代码一般应调入 SDRAM 中运行，以提高系统的运行速度，同时，系统及用户堆栈、运行数据也都放在 SDRAM 中。